EXT: Shop System

Extension Key: **tt_products**

Copyright 2005-2006,Franz Holzinger <kontakt@fholzinger.com>

Dieses Dokument ist unter der Open Content Lizenz veröffentlicht,
die unter http://www.opencontent.org/opl.shtml erhältlich ist.

Der Inhalt dieses Dokuments bezieht sich auf TYPO3
- ein GNU/GPL CMS/Framework, das unter www.typo3.com erhältlich ist.

Table of Contents

Einleitung ... **2**
- Version ... 2
- Übersetzungen ... 2
- Upgrade ... 2
- Wofür ist das gut? ... 2
- Bildschirmfoto ... 3
- Entwickler ... 3
- Sponsoren ... 3
- Unterstützung ... 3
- Support ... 4

Benutzerhandbuch ... **5**
- Hinweis: ... 5
- Mini-Warenkorb: ... 5
- Berechnungsskript: ... 6
- Rabatt: ... 6
- Link für die Suche nach Produkten der letzten X Tage ... 6
- Angebote und Highlights ... 6
- AGB Allgemeine Geschäftsbedingungen ... 6
- MEMO-Vormerk Seite ... 6
- Freundschaftswerbung ... 6
- Gutscheinpunkte System ... 7
- Geschenk Gutscheine ... 7
- Kauf in einer Kiste ... 7
- Gebrauch der Produkt Artikel (Varianten mit eigenen Preisen) ... 7

Administration ... **9**
- Installation ... 9
- Behandlung der Kategorien ... 9
- Wichtige Hinweise ... 9
- Template Marker ... 9
- Automatisches Erzeugen von Frontend Benutzern ... 12
- Produkt Eigenschaften: ... 12

Konfiguration ... **15**
- FAQ ... 15
- Dateien ... 15
- Beschreibung ... 15
- Referenz ... 15
- class.tx_ttproducts.php properties ... 15
- CSS Konfiguration ... 24
- Konfiguration der Artikel, Produkte, Kategorien, Seiten und Bilder ... 25
- Formular Einstellungen ... 26
- Konfiguration für Bezahlung und Versand ... 27
- Pricecalc, discountprice und creditpoints Konfiguration ... 30
- payment_DIBS.php ... 32

Tutorial ... **33**

Bekannte Probleme ... **36**
- Prüfliste: ... 36
- Allgemein: ... 36

To-Do Liste ... **37**

Changelog ... **37**

Einleitung

Bevor Sie mit dem Shop beginnen, sollten Sie das deutsche Tutorial durchgearbeitet haben.

Version

Dieses Dokument gilt für tt_products Version 2.5.1.

Übersetzungen

Die Englische Übersetzung dieses Dokuments wird mit tt_products ausgeliefert (Pfad tt_products/doc). Eine französische Übersetzung doc_ttproducts_fr ist in Bearbeitung und sucht noch Helfer.

Upgrade

Wenn Sie ein Upgrade von einer früheren Version machen wollen, dann müssen Sie unbedingt die wichtigen Hinweise unter 'Administration' befolgen.

Ab 1. Dezember 2006 kann PHP5 für die Installation neuer Versionen Voraussetzung sein.

Wofür ist das gut?

Die Typo3 Shop Extension gibt Ihnen die Möglichkeit zu...

- Produkte mit mehreren Bildern, Details und in mehreren Sprachen aufzulisten
- Einkaufskorb
- “Bezahlen” Seite – Hier wird die Bestellung angezeigt und kann überprüft werden, bevor die Artikel endgültig bestellt werden/die Bestellung endgültig versandt wird
- für Sponsoren: Bezahlsysteme über die Payment Library – Transaction Central und Paypal
- Nachverfolgen des Zustands einer Kundenbestellung (Tracking)
- Automatische Erzeugung einer Rechnung und eines Lieferscheins
- Verschiedene Steuersätze pro Artikel, für die Zustellung und die Bezahlung
- einfache Lagerverwaltung
- Versenden einer CSV Datei für jede Bestellung an den Shop Administrator (2 auswählbare Dateiformate)
- E-Mail-Anhänge für Bestätigungs Emails (zum Beispiel AGB = Allgemeine Geschäftsbedingungen)
- auswählbare Artikelvarianten (Farben, Größen, Staffelungen)
- Zwang zum Akzeptieren der Allgemeinen Geschäftsbedingungen (AGB) durch den Kunden über eine Checkbox
- Angebote, Highlights and neu hinzugefügte Artikel
- Spezialanfertigung, Gewicht und Sperrgut (kann für die Berechnung der Versandkosten verwendet werden)
- Möglichkeit, die Bezahlungsmethoden auf besondere Benutzergruppen einzuschränken
- Automatische Erzeugung eines Frontend Benutzers bei der ersten Bestellung
- Vormerkliste für Artikel – eingeloggter Benutzer kann mit der Memo-Funktion Artikel vormerken/speichern
- Rabatt in Prozent pro Benutzer
- Methoden der Preisberechnung mit Rabatten für Wiederverkäufer/Gruppen
- Anzeige der Bestellungen
- Gutscheinpunkte System: Kunden können mit jeder Bestellung Gutscheinpunkte sammeln. Die gespeicherten Gutscheinpunkte kann der Kunde bei einer neuen Bestellung einlösen . Auf Ihren Wunsch können mit diesen Gutscheinen auch nur bestimmte Produkte gekauft werden.
- Freundschaftswerbung: Wenn ein Kunde bei seiner Registrierung angibt, daß er von einem anderen (bereits existierenden) Kunden angeworben wurde, dann erhält der Anwerber eine Gutschrift in Gutscheinpunkten. Der neue Kunde erhält bei seiner ersten Bestellung einen Rabatt.
- Geschenkgutscheine: Benutzer können Geschenkgutscheine kaufen und diese per Email an ihre Geschäftspartner senden. Wenn sich dieser Geschäftspartner im Frontend anmeldet, dann kann er seinen Geschenkgutschein in Gutscheinpunkte umwandeln und bereits bei der ersten Bestellung einsetzen.

Bildschirmfoto

Entwickler

- Kasper Skårhøj: Entwickler bis 1.2.7 / TYPO3 3.8
- René Fritz: Entwickler bis 1.2.7 / TYPO3 3.8
- Franz Holzinger (Projekt Leiter): Preisberechnung, Diskont Preis, Geschenkgutscheine, Email Tabelle für Benachrichtigung, Fehlerkorrekturen, Rechnung, Lieferschein, mehrspaltige Listenansicht, Artikel Tabelle, mehrsprachige Tabellen
- Klaus Zierer (Leiter von zk_products): mehrere Größen und Farben, Vormerkliste, AGB, CSV, automatische Registrierung der Frontend Benutzer, Seiten als Kategorien, ein beispielhaftes deutsches Template.
- Milosz Klosowicz (Leiter von mkl_products): verschiedene Währungen, Flexform
- Els Verberne: Gutscheinpunkte und Freundschaftswerbung System, Liste der Bestellungen

Sponsoren

- Die Entwicklung der Geschenkgutscheine wurde von Franz Holzinger durchgeführt. Es wurde von BENDOO e-work solutions aus den Niederlanden (http://www.bendoo.nl) gesponsert. Mehr Information erhalten Sie direkt bei BENDOO: hiddink@bendoo.com
- Der Verkauf von Artikeln ohne gemeinsame Attribute mit Produkten, die Kategorie Auswahl-Box, das Benennen der Bilder aus Teilen von Tabellenfeldern und der DAM Tabelle, und das Setup des Versands über die static_countries Tabelle wurde von Franz Holzinger durchgeführt. Es wurde von pr!int - public relations & internet, Spiegelberg / Deutschland (http://www.pr-int.biz) gesponsert. Nehmen Sie mit pr!int unter office@pr-int.biz Kontakt auf.

Unterstützung

tt_products ist ein Open Source Shop System für TYPO3. Sie dürfen es verwenden, ohne eine Lizenzgebühr dafür bezahlen zu müssen. Aber dennoch steckt eine Menge Arbeit dahinter, damit dieses Shop System auch einwandfrei funktioniert, damit aufgetretene Fehler möglichst bald korrigiert werden, damit die Dokumentation gepflegt wird, damit es Online Support dafür gibt und schließlich, daß es weiterentwickelt wird.

Getane Arbeit von 2005:

Mehrsprachige Tabellen, Übersetzung der Template Dateien, Integration von zk_products und Teilen von mkl_products, eine bessere Dokumentation und ein Deutsches Tutorial, Verbesserungen und Korrekturen am Code, jede Menge an Support auf der tt_products Mailingliste und in verschiedenen TYPO3 Foren.

Am Quelltext im CVS Repository unter http://svn.sourceforge.net/viewcvs.cgi/typo3xdev/tx_ttproducts/trunk/ können Sie die Beiträge der Entwickler mitverfolgen.

Für die Weiterentwicklung wäre es wichtig, daß Sie vor allem die folgenden Personen kontaktieren, um Ihnen eine Anerkennung für ihre Arbeit zu geben. Sie bieten Ihnen auch die Entwicklung neuer Features zu tt_products an.

- Franz Holzinger <kontakt@fholzinger.com>:
  http://fholzinger.com Tel +49 221 7592169
- Jens Schmietendorf <jens@smedja.com>:
  http://webb.smedja.com, Tel.: +46 471 22102
- Els Verberne <verberne@bendoo.nl>:
  BENDOO e-work solutions http://www.bendoo.nl, Tel. +31 315 244525

Wenn Sie selber Programmierer sind und Ihre eigenen Erweiterungen zu tt_products geschrieben haben, so sind Sie eingeladen, den Code an Franz Holzinger zu senden. Wenn der Code von allgemeinem Interesse für andere ist, dann könnte er ein paar Korrekturen anbringen und es dann mit der nächsten Version von tt_products veröffentlichen. Damit würde Ihre heutige Entwicklung auch bei späteren Updates automatisch mit dabei sein.

Support

Sie können unter http://fholzinger.com/index.php?id=162 Support dazu erhalten, oder auf anderen Seiten im Internet.

Benutzerhandbuch

Hinweis:

Der Pfad zu den Dateien wurde von 'pi' auf 'pi1' umbenannt, damit tt_products jetzt gleich aufgebaut ist wie andere TYPO3 Extensionen. Daher müssen Sie nun Ihre Pfade ändern, wenn Sie diese im TypoScript Setup von älteren Versionen von tt_products verwendet haben.

Wenn Sie die Extension neu installieren oder updaten: Bitte beachten Sie die Depencies im TER (Online Repository des Extension Managers). Es muss die Table Library installiert sein.

Sie müssen das Feld 'Am Lager (St.)' in den Eigenschaften des Produktes ausfüllen. Ansonsten wird bei der Ausgabe im Frontend kein Artiekel angezeigt, da der Wert dann auf 0 gesetzt ist. Null Artikel sind eben nicht vorhanden.

Negative Werte haben eine besondere Bedeutung: -1 bedeutet, dass eine Ware immer 'Am Lager' ist. Das Produkt ist damit immer lieferbar.

Vor der Eingabe der Produktpreise haben Sie die Möglichkeit zu entscheiden, ob die Preise mit oder ohne Steuern eingegeben werden (siehe TAXincluded). Alle Berechnungen und Konfigurationen werden die Preisangaben verwenden, mit denen sie in der Tabelle tt_products gespeichert sind. Sie sollten jedoch am besten die Preise inkl. Mehrwertsteuer in die Datenbank eintragen. Das erleichter es Ihnen, die Preise gleich so eingeben zu können, wie sie auch der Kunde sehen wird. Diese Preise sollten schließlich den Konventionen der Preisgestaltung wie € 1,99 haben und nicht € 1,72.

Wenn Sie mit Ihrem neuen Shop beginnen, dann verwenden Sie ein Beispiel-Template aus dem Verzeichnis tt_products/template, das Template example_template_bill_de.tmpl Sie müssen nur noch die Seiten-Id in den Links (hinter 'id=...') anpassen, damit hier die Seiten-Id des Warenkorbs ihrer Seite steht.

Der Marker ###DOMAIN### erspart Ihnen die Eingabe Ihrer URL. Das ist sinnvoll, wenn z. B. ein neuer Host verwendet wird. Im setup des Shoptemplates muss die Domainstellung vorgenommen werden: plugin.domain = meinedomain.de

Wenn Sie Einträge in die Variant-Felder (Größe, Farbe ...) vornehmen, müssen Sie u. U. die selectSize und selectColors entsprechend auf 0 oder 1 setzen. Ansonsten werden in der Produktliste die Anzahlen der Artikel falsch eingetragen.

Viele Anpassungen können über den Constant Editor vorgenommen werden. Das meiste sollte besser direkt in die Felder Setup und Constants der Templates für die Shop-Seite eingetragen werden.

tt_products basiert auf den Extensionen Table Library (table) und FE/BE Library (fh_library). Wenn Sie also ein Update von tt_products durchführen, dann müssen Sie auch gleichzeitig ein Update dieser beiden Extensionen durchführen.

Mini-Warenkorb:

Neben dem “normalen” Warenkorb, der eine komplette Liste der eingegebenen Artikel in einer gewissen Sortierreihenfolge anzeigt, gibt es auch einen "Mini-Warenkorb", der nur die Anzahl der Produkte im Warenkorb und den Gesamtpreis anzeigt.

Dieser zeigt Ihnen während der Verweildauer im Shop ständig aktualisiert den Inhalt Ihres Warenkorbes oberhalb der normalen Ansicht (oder wo Sie wünschen) an. Die Stelle, an der der “Mini-Warenkorb” angezeigt werden soll, kann über Typoscript gesteuert werden. Um diesen Mini-Warenkorb verwenden zu können, fügen Sie ein neues Seiteninhaltselement "Plugin: Produkte" ein und tragen in das CODE-Feld OVERVIEW ein. Verwenden Sie die folgenden Markierungen in Ihrem tt_products Template um den Mini-Warenkorb zu aktivieren und anzupassen:

Code Listing:

```
<!-- ###BASKET_OVERVIEW_TEMPLATE### begin -->
  <div class="shop_minibasket">
     <div class="shop_minibasket_image"><!--###LINK_BASKET###-->###IMAGE_BASKET###<!--
###LINK_BASKET###--></div>
    <p>###NUMBER_GOODSTOTAL### Produkte (EUR ###PRICE_GOODSTOTAL_TAX###.-)</p>
<!-- ###BASKET_OVERVIEW_TEMPLATE### end -->


<!-- ###BASKET_OVERVIEW_EMPTY### begin -->
  <p>Keine Produkte im Warenkorb.</p>
<!-- ###BASKET_OVERVIEW_EMPTY### end -->
   </div>
```

Berechnungsskript:

Wenn Sie Ihr eigenes Berechnungsskript schreiben und verwenden wollen, dann sollten Sie immer die Preise inkl. Mehrwertsteuer eingeben. Die priceTax und priceNoTax Variablen werden Ende August 2008 gelöscht werden, weil sie aus der price Variable und den Konstanten TAXincluded und TAXpercentage berechnet werden können. Sie können die global PHP-Variablen $this->calculatedArray und $this->itemArray verwenden, um Ihre eigenen Preisberechnungen durchzuführen.

Rabatt:

Es gibt mehere Methoden um einen Rabatt zu erhalten.

1. Verwenden des Rabatt Feldes eines Frontend Benutzers und Eingabe der Prozente für den Discount
2. Verwenden des Feldes price2 für besondere Feuser Gruppen und andere Dinge. Sie können das in TypoScript über das

IF-statement setzen.

3. Verwenden der Discountprice Berechnung. Der Rabatt hängt von der Gesamtanzahl der Produkte mit einem besonderen Preis ab. Dieser Rabatt wird für die Frontend-Benutzer der Gruppe 'discountGroupName' gewährt.

Link für die Suche nach Produkten der letzten X Tage

Das ist praktisch für Seiten “Was ist neu?”. Erzeugen Sie einfach einen Link zur SEARCH Seite und geben Sie "newitemdays" als Parameter an. http://www.mytypo3shop.com/?id=8&newitemdays=10 Die Suche wird alle Produkte der letzten 7 Tage liefern. Sie müssen den Plugin Code LISTNEWITEMS eintragen. Sie können stattdessen auch die Konstante "newItemDays" setzen.

Angebote und Highlights

Sie müssen den Plugin Code LISTOFFERS zur Anzeige der als 'Angebot' markierten Artikel verwenden. Und mit LISTHIGHLIGHTS erhalten Sie eine Liste besonderer Artikel (Highlights).

AGB Allgemeine Geschäftsbedingungen

Sie müssen die Allgemeinen Geschäftsbedingungen akzeptieren bevor Sie eine Bestellung machen und mit dem Bestellvorgang fortfahren können. Zu diesem Zweck kann eine Seite als "PIDagb" definiert werden. Die AGBs müssen hier angezeigt werden. Sie können das auch als das Ziel eines Links über "agbtarget" setzen.

Ein Dokument (z.B. eine PDF Datei) auf dem Server, welche die AGB enthält, kann über "AGBattachment" gesetzt werden. Diese wird als Anhang an eine Email mit der Bestellbestätigung an den Kunden geschickt. Die Shop Administration wird diese Datei nicht erhalten. (Sie kann stattdessen eine CSV Datei mit den Bestelldaten erhalten, falls gewünscht).

MEMO-Vormerk Seite

Der eingeloggte Benutzer kann Artikel zu einer Vormerk-Seite hinzufügen. Diese Memo Seite wird in der fe_user Tabelle gespeichert. Erzeugen Sie einfach einen neue Seite mit dem Shop Plugin als Inhalt und geben Sie den Code "MEMO" ein. Sie müssen auch den Anfangspunkt setzen, wie auf den SEARCH Seiten.

Freundschaftswerbung

Ein interessierter Kunde erhält per Email eine Einladung ein Shop zu besuchen und dort Waren einzukaufen. Dieser Einladung ist ein einmaliger Code beigelegt, der einer gewissen Anzahl von Gutscheinpunkten entspricht (Gutschrift). Wenn sie einen Kauf tätigen, können Sie diese Gutschrift einlösen, indem diese durch Gutscheinpunkte für den Frontend Benutzer ersetzt werden.

Wenn ein Kunde bei seiner Registrierung angibt, daß er von einem anderen Kunden angeworben worden ist, dann erhält jener Kunde Gutscheinpunkte. Der neue Kunde erhält bei seiner ersten Bestellung einen Rabatt.

Jemand (sagen wir Benutzer 1) kann einen Gutschrift Code eingeben, wenn er eine Bestellung tätigt. Dann wird Benutzer 1 5 Euro Rabatt erhalten. Dieser Gutschrift Code besteht aus dem Benutzernamen von jemand anderem (wwelche die Email-Adresse ist), sagen wir Benutzer 2. So erhält Benutzer 2 insgesamt 5 Gutscheinpunkte. Jeder Benutzer kann nur einmal einen Gutschrif Code aus einer Freundschaftswerbung verwenden. In der fe_users Tabelle gibt es dafür ein Feld "tt_products_vouchercode", das den verwendeten Gutschrift Code (Email von Benutzer 2) enthält.

Gutscheinpunkte System

Wenn ein Kunde eine gewisse Anzahl an Waren gekauft hat, dann gewinnt er dadurch Gutscheinpunkte, die auf seinem Benutzerkonto gespeichert werden. Zu einem späteren Zeitpunkt, wenn er weitere Waren kauft, kann er diese Gutscheinpunkte ein Waren einlösen und beim Kauf einen besonderen Rabatt lukrieren. Die Kunden können bei jedem Kauf Gutscheinpunkte sammeln.

```
creditpoints {
 pricefactor = 0.5
 10.type = price
 10.prod.1   = 0.02
 10.prod.101 = 0.04
 10.prod.501 = 0.06
}
```

Geschenk Gutscheine

Ein Shop Betreiber kann z.B. Gutscheine für Weinflaschen als “elektronisches Geschenk” verkaufen. Ich bestelle einen Gutschein, der z.B. 50 Gutscheinpunkten entspricht (=25 Euros), sende ihn Ihnen zu. it to you. Wenn Sie diesen Geschenkgutschein einlösen wollen, dann müssen Sie sich als Kunde des Shop registrieren und den Code auf diesem Gutschein angeben. Danach erhalten Sie die 50 Gutscheinpunkte Ihrem Konto gutgeschrieben. Mit diesen Gutscheinpunkten können Sie bestimmte Produkte einkaufen. Jemand kann z.B.: 3 Gutscheine zu 25 Gutscheinpunkten 5 Gutscheine zu 50 Gutscheinpunkten 8 Gutscheine zu 75 Gutscheinpunkten ...alle mit derselben Bestellung kaufen.

Jeder kann zum Beispiel

- 3 Gutscheine à 25 Gutscheinpunkte
- 5 Gutscheine à 50 Gutscheinpunkte
- 8 Gutscheine à 75 Gutscheinpunkte

...alles mit einer einzigen Bestellung kaufen.

Kauf in einer Kiste

In einem Shop wird Wein verkauft. Normalerweise verkaufen Sie eine Kiste von Weinflaschen (Eine Kiste kann 6 oder 12 Artikel enthalten). Der zusätzliche Service "Kauf-einer-Kiste" gibt dem Shop Kunden die Möglichkeit, sich seine eigene Kiste von 6 Flaschen einzupacken: z.B. 2 rote, 2 weiße, 2 rosé .

Unter dem Kauf-einer-Kiste Konzept wird die besondere Bearbeitung nach den Flaschen und Kisten berechnet. So wird die Bearbeitung pro Flasche mit 1,50 Euro/Flasche berechnet, die Extra Bearbeitung für 2 Kisten würde 2 x 6 x 1,50 = 18 Euro betragen.

Gebrauch der Produkt Artikel (Varianten mit eigenen Preisen)

Hier wird beschrieben, wie verschiedene Größen eines Produktes zu verschiedenen Preisen verkauft werden können. Das selbe Verfahren können Sie auch mit Farben und den anderen Varianten anwenden. Achten Sie darauf, daß ' Zweite Optionspalette anzeigen' abgehakt worden ist, sonst werden einige der Felder nicht angezeigt.

1. Im TSSetup für das Template geben Sie ein

   ```
   plugin.tt_products.useArticles = 1
   plugin.tt_products.selectSize = 1
   ```

   Wenn sie Spalten für die Listenanzeige der Produkte verwenden, wie im Bananaguard Beispiel, dann benötigen Sie auch

   ```
   plugin.tt_products.displayBasketColumns.selectSize = 1
   ```

   Das teilt dem Shop mit, daß die Daten aus der Produkte Artikel Tabelle gelesen werden und daß eine Auswahbox dargestellt wird, wenn Varianten verwendet werden.

2. Geben Sie das Vaterprodukt an.
   Das ist die erste Information, die in der Frontend Liste und in Einzelansichten angezeigt wird, z.B. in der Beschreibung und über die Bilder. Erzeugen Sie einen neuen Eintrag im Web > Liste Modul. Wählen Sie das Produkt aus..
   Sie müssen folgende Felder ausfüllen.

   - Titel
   - auf Lager (muß mindestens 1 betragen, damit es im Frontend angezeigt wird)
   - Kategorie (falls benötigt)
   - Preis (das sollte der niedrigste Preis für die billigste Größe sein – das entspricht dem billisten Produkt Artikel, den wir als nächstes eingeben werden)
   - Größe (dasselbe gilt für Farbe/Beschreibung/Staffelung falls benötigt)
     Es ist unbedingt notwendig, um im Frontend die Möglichkeit zu haben, eine Produktvariante auswählen zu können, daß alle Größenvarianten in eine durch Strichpunkt getrennten Liste eingegeben werden, z.B.: S;M;X;XL . Diese Artikel wollen wir als nächste eingeben.
   - Bemerkung
   - Bild
   - Datenblatt (falls benötigt).

3. Hinzufügen der Produkt Varianten.
   Hier erzeugen wird die verschiedenen Größen für das Vaterprodukt. Die Daten dieses Produktes können nun über den Artikel überschrieben werden. Das wird nur benötigt, wenn zumindest der Preis anders sein soll. Jede Produkt Variante muß einen entsprechenden Eintrag im Größenfeld (oder Farbe etc.) des Vater Produktes haben.

   - Legen Sie einen neuen Datensatz im Web > Liste Modul an.
   - Wählen Sie Produkt Artikel aus.
   - Füllen Sie das Titelfeld (das wird im Warenkorb angezeigt) aus und
     - Am Lager,
     - Preis

       Wenn diese Variante die billigste Version ist, dann soll es derselben Preis wie das Vaterprodukt haben.
     - Produkt

Es ist wichtig, daß Sie das Vaterprodukt im Elementbrowser auswählen. Ansonsten wird der Preis im Warenkorb nicht geändert, wenn die Größe des Produktes verändert wird.

- Größe (oder Farbe, Beschreibung, Staffelung falls benötigt)
  Das muß einer der Größen entsprechen, die Im Vaterprodukt aufgelistet worden sind. z.B. XL.
  Wenn die Schreibweise nicht identisch ist, wird es nicht funktionieren.

4. Anpassungen im HTML Template
   Das ist als letzes zu machen.

   - Sie müssen neue Felder in die ###ITEM_LIST_TEMPLATE### und ###ITEM_SINGLE_DISPLAY### Bereiche einfügen.
   - Der folgende HTML Code muß für die Anzeige der Auswahlbox der Varianten eingefügt werden, und zwar innerhalb der Form-Tags. Außerdem wird jede Information innherhalb von ###display_variant1### oder ###display_variant2### Kommentaren nicht angezeigt, wenn diese Variante nicht verwendet wird. Das ist sehr nützlich, wenn das Wort 'Ab' vor dem ###PRICE_TAX### Wert innerhalb der ###display_variant2### Kommentare eingefügt wird. So wird z.B. 'Ab $200' wenn die billigste Größe €200 kosten soll (deshalb wird der niedrigeste Preis im Vaterprodukt eingetragen. - standardmäßig wird der Preis des Vaterprodukts in der Listen- und Einzelansicht angezeigt. Der Preis der Variante wird erst im Warenkorb sichtbar, wenn eine bestimmt Größe (oder Farbe etc.) ausgewählt worden ist. Wenn keine Varianten zur Auswahl stehen, dann wird das Wort 'Ab' auch nicht angezeigt.

     Code Beispiel:

```
<!-- ###display_variant1### -->
Farbe:><SELECT name="###FIELD_COLOR_NAME###" rows="1">###PRODUCT_COLOR###</SELECT>
<!-- ###display_variant1### --><!-- ###display_variant2### -->
Gr&ouml;e:><SELECT name="###FIELD_SIZE_NAME###" rows="1">###PRODUCT_SIZE###</SELECT>
<!-- ###display_variant2### -->
```

Administration

Im Normalfall werden die Kategorien über die Kategorietabelle und ihre Erweiterungen angezeigt. Um das zu nutzen, muß

$TYPO3_CONF_VARS['EXTCONF']['tt_products']['pageAsCategory'] = 0

eingestellt sein. Andernfalls bestimmen die Seite die Kategorie und die Kategorie ist die Unterkategorie (=1) oder die Kategorie wird nicht benutzt sondern komplett durch die Seite ersetzt (=2).

Aktivieren Sie schon jetzt die Flexforms im Backup. Dabei wird die Eingabe des Codes in das CODE Feld ersetzt durch eine graphische Oberfläche. Allerdings müssen Sie alle Codes neu eingeben.

$TYPO3_CONF_VARS['EXTCONF']['tt_products']['useFlexforms'] = 1

Installation

Installieren Sie die Extension über den Extension Manager. Wenn Sie bereits eine ältere Version von tt_products verwenden, dann ist ist in der “global” Umgebung (typo3/ext/) installiert. Wenn Sie eine neue Version ausprobieren wollen, dann sollten Sie diese unter “lokaler” Ordner (typo3conf/ext/) installieren, damit Sie die alte Version nicht dabei löschen.

Deinstallieren Sie die Extension über den Extension Manager. Danach laden Sie die eine Version Ihrer Wahl mit dem Extension Manager vom Online Repository herunter. Dann verwenden Sie im Extension Manager "Available extensions to install" und fügen das Shop System wieder hinzu.

Diese Extension läuft am besten, wenn Sie zusammen mit den Extensionen static_info_tables, sr_feuser_register, rtehtmlarea und conf_userts verwendet wird.

Behandlung der Kategorien

Es gibt mehrere Möglichkeiten ein Shop aufzubauen. Der normale Weg läuft über die Erzeugung von Sysfolder für die Produkte. Für den Aufbau eines größeren Shops mit TYPO3 wäre das etwas unpraktisch.

Wenn Sie größere Shops mit hierarchischen Kategorien anlegen wollen, dann sollten Sie die mbi_products_categories Extension und vielleicht auch nsb_cat2menu installieren.

Beispiel of tt_products/ext_localconf.php:

```
$TYPO3_CONF_VARS[’EXTCONF’][’tt_products’][’pageAsCategory’] = 0;
```

Wichtige Hinweise

Das Shop ist nach Möglichkeit abwärts kompatibel zur früheren Version tt_products 1.2.7. Aber die Template Datei benötigt einige kleine Anpassungen. Am besten Sie vergleichen Ihr Template mit den Beispiel-Templates, die mit tt_products mitgeliefert werden. Im ###BASKET_TEMPLATE### müssen Sie die Zeilen

```
<input type="hidden" name="mode_update" value="1">
<input type="Submit" name="products_update" value="update basket">
```

haben.

Sie müssen im BASKET_TEMPLATE den Marker ###FIELD_NAME### auf ###FIELD_NAME_BASKET### umbenennen.

Das ist kompatibel zu zk_products 1.3.2.

Einige der TypoScript Einstellungen funktionieren nicht über das Constants Feld und müssen im Setup Feld eingetragen werden. Nur die Konfigurationen, die im Konstanteneditor oder in der Datei static/old_style/constants.txt vorkommen, sind gültige Konstantenwerte. Sie müssen das 'Shop System Old Style' unter 'statische Template-Datensätze' in Ihr TYPO3 Template einbinden.

Ein negativer Wert eines Produktes bei 'auf Lager' hat keine Sonderfunktion mehr. Dies wird nun durch eine Checkbox 'immer auf Lager' ersetzt.

Ändern Sie die früheren Eingabefelder der Suche in

<INPUT size="30" maxlength="100" type="text" name="sword" value="###SWORD###">

Die Marker ###SWORDS### müssen in ###SWORD### geändert werden, ebenso der Name auf 'sword'.

Template Dateien

In den Template Dateien Ihres Shop müssen Sie die folgenden Korrekturen machen, damit Sie weiterhin verwendet werden können:

<input type="hidden" name="mode_update" value="1">

<input type="Submit" name="products_update" value="update basket">

Stellen Sie diese FORM-Attributes in Ihr BASKET_TEMPLATE.

Template Marker

Die folgenden Template Marker werden für die Template-Datei verwendet. Sie müssen ### vorher und danach eingeben. Es gibt noch einige andere Marker, die Sie bitte den Beispiel-Templates entnehmen.

Bereichsmarker:

Markierung:	Beschreibung:	Bereich:
BASKET_TEMPLATE		top
BASKET_INFO_TEMPLATE	Das wird normalerweise dazu verwendet, damit die Kunden ihre Adreßdaten außerhalb des Warenkorbs eingeben. Es hat dieselben Eigenschaften wie BASKET_TEMPLATE.	top
BASKET_ORDERCONFIRMATION_TEMPLATE	Die letzte Seite, nachdem die Bestellung verarbeitet worden ist. Wenn PIDthanks gesetzt ist, wird das nicht für die Anzeige benutzt. Aber diese Teile werden immer für die Benachrichtigungsemails verwendet, wenn HTML-Emails eingestellt sind (orderEmail_htmlmail=1). siehe: BASKET_ORDERTHANKS_TEMPLATE	
BASKET_ORDERCONFIRMATION_NOSAVE_TEMPLATE	Diese HTML-Daten werden angezeigt, nachdem die Bestellung in der Datenbank gespeichert worden ist.	
BASKET_ORDERTHANKS_TEMPLATE	Anzeige einer 'Danke schön' Seite, wenn PIDthanks gesetzt ist. Für den Versand von HTML-Emails wird das aber nicht verwendet. siehe BASKET_ORDERCONFIRMATION_TEMPLATE	
BASKET_OVERVIEW_EMPTY	Meldung 'Ihr Warenkorb ist noch leer' für OVERVIEW	
BASKET_OVERVIEW_TEMPLATE		
BASKET_PAYMENT_TEMPLATE		
BASKET_REQUIRED_INFO_MISSING		
BASKET_TEMPLATE_EMPTY	Meldung 'Ihr Warenkorb ist noch leer'	
BASKET_TEMPLATE_NOT_LOGGED_IN	Fehlermeldung, daß sich der Benutzer nicht eingeloggt hat	
BASKET_TEMPLATE_INVALID_GIFT_UNIQUE_ID	Fehlermeldung, daß eine falsche Kennnummer für ein Geschenk-Zertifikat eingegeben worden ist	
BILL_TEMPLATE	zur Erzeugung der Rechnung	
DELIVERY_TEMPLATE	Erzeugung der Datei des Lieferscheins	
EMAIL_PLAINTEXT_TEMPLATE	Text für die Email Benachrichtigung Die erste Zeile wird für das Betreff Feld verwendet.	
EMAIL_NEWUSER_TEMPLATE	Email nach der Erzeugung eines Frontend Benutzers	
ITEM_LIST_TEMPLATE	Auflistung der Produkte auf der Startsetite LIST	
ITEM_LIST_GIFTS_TEMPLATE	Auflistung der Produkte Geschenk-Gutscheine	
ITEM_SEARCH		
ITEM_SEARCH_EMPTY		
ITEM_SINGLE_DISPLAY		
ITEM_SINGLE_DISPLAY_GIFT	Einzelansicht für Geschenkgutscheine	
ITEM_SINGLE_DISPLAY_NOT_IN_STOCK	Einzelansicht für Artikel, die nicht auf Lager sind	
ITEM_SINGLE_DISPLAY_RECORDINSERT	siehe displayCurrentRecord: generieren des $cObj->data	
MEMO_TEMPLATE		
MEMO_NOT_LOGGED_IN	Fehlermeldung, daß sich der Benutzer für MEMO nicht eingeloggt hat	
ORDERS_LIST_TEMPLATE	Anzeige der Liste der Bestellungen	
TRACKING_EMAIL_GIFTNOTIFY_TEMPLATE	Benachrichtigung an den Empfänger eines Geschenkgutscheins in der Verwaltung der Bestellungen	

Markierung:	Beschreibung:	Bereich:
TRACKING_ENTER_NUMBER		
TRACKING_WRONG_NUMBER		

Einzelmarker:

Marker:	Typ:	Beschreibung:	Bereich:
BROWSE_LINKS		Zum Durchblättern über mehrere Seiten in der Anzeige	
DELIVERYCOSTS	value	Summe der Lieferkosten und Kosten für die Bezahlung	
DELIVERY_... NOTE NOTE_DISPLAY DESIRED_DATE	value	Siehe auch PERSON_... Eingabefeld Bemerkung zur Bestellung Ausgabe Bemerkung mit für den Zeilenumbruch gewünschtes Lieferdatum	BASKET_INFO_TEMPLATE
EXTERNAL_COBJECT	value	zusätzliches Cobject	
FIELD_NAME	value in form	Name eines Feldes in einem HTML Formular	
FIELD_NAME_BASKET	value in form	Die Daten des Warenkorbs in verschlüsselter Form.	
GC1, GC2, GC3	value	globale Farben	all
GW1B, GW2B, GW1E, GW2E	wrap	globale Wraps (HTML-Umrandungen)	all
PRICE_TAX PRICE_NO_TAX PRICE2_TAX PRICE2_NO_TAX	value	price/price2 of the article including or excluding the tax	ITEM_SINGLE
PRICE_TOTAL_TAX PRICE_TOTAL_NO_TAX	value	total sum of all items with payment and shipping costs	all
PRICE_GOODSTOTAL_TAX PRICE_GOODSTOTAL_NO_TAX PRICE2_GOODSTOTAL_TAX PRICE2_GOODSTOTAL_NO_TAX	value	total sum of all items for price or price2	all
LINK_DATASHEET		Link auf die Datenblatt Datei in uploads/tx_ttproducts/datasheet	
ORDER_STATUS_TIME, ORDER_STATUS, ORDER_STATUS_INFO, ORDER_STATUS_COMMENT	value	Daten zur Bestellung	TRACKING_DISPLAY_INFO
PERSON_... NAME, ADDRESS, TELEPHONE, FAX, EMAIL, COMPANY, CITY, ZIP, STATE, COUNTRY	value	Felder für die Adreßinformation ein Präfix muß vorangestellt sein PERSON_... der Kunde der Bestellung DELIVERY_... der Empfänger der Bestellung	
PID_TRACKING	value	Die Seitennummer für den Rückverfolgung über den Status der Bestellung	
STATUS_CODE_60	wrap	wird verwendet, damit das Zusenden von Emails an den Empfänger eines Geschenkgutscheins erst nach der Bezahlung erscheint	TRACKING_DISPLAY_INFO
STATUS_OPTIONS	value	Auswahlmenü für Zustände in der Bestellungsverwaltung	ADMIN_CONTROL inside TRACKING_DISPLAY_INFO
SHOPADMIN_EMAIL	value	Email-Adresse des Shop-Adminstrators	all

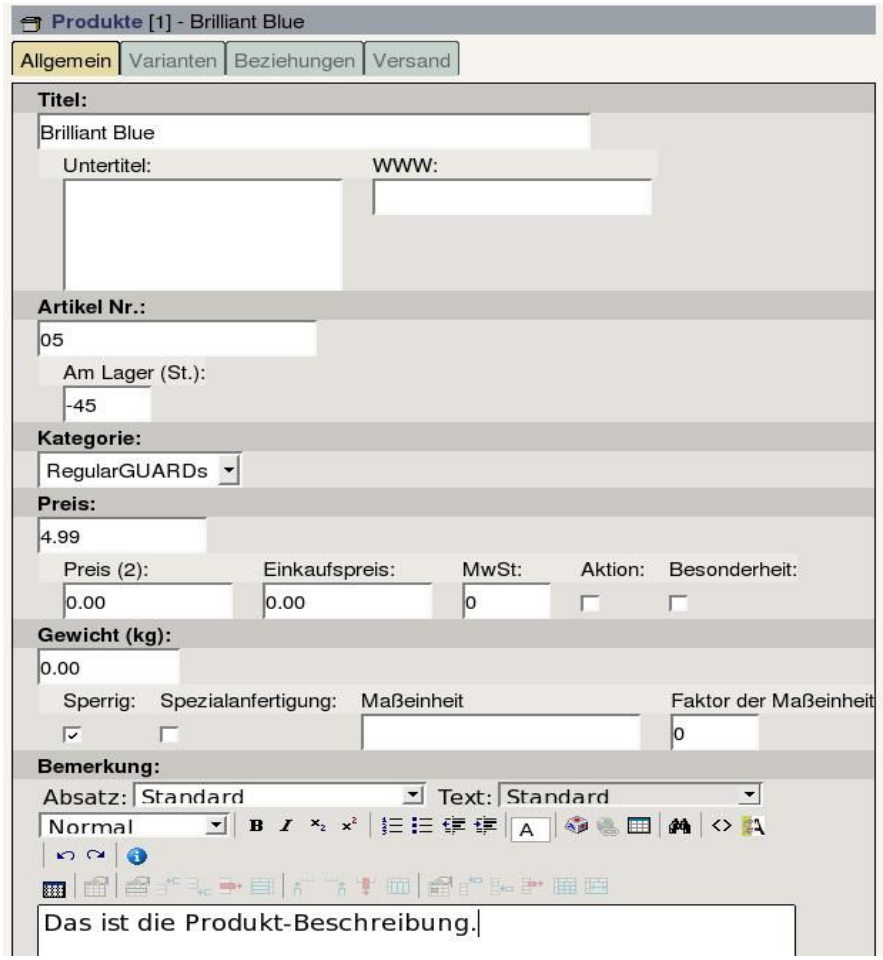
Automatisches Erzeugen von Frontend Benutzern

Es ist möglich nach jeder Bestellung automatisch Frontend Benutzer anzulegen. Sie müssen "createUsers" auf "1" setzen, die PID des Sysfolder als PIDuserFolder im Setup Feld eintragen. Dann müssen Sie die memberOfGroup auf die ID Ihrer Frontend Benutzergruppe setzen. Nach der ersten Bestellung erhält der Kunde eine Email mit seinen Zugangsdaten. Diese Email wird den Namen des Accounts enthalten, welcher ident mit der Email Adresse ist, und sein automatisch erzeugtes Paßwort.

Wenn Sie kein einzelnes Textfeld für die Adresse haben wollen, sondern eine für die Straßenbezeichnung und Hausnummer, und zusätzliche Felder für PLZ, Stadt und Land, dann müssen Sie die Template-Datei entsprechend anpassen. Wenn Sie static_info_tables installieren und “useStaticInfoCountry=1” setzen, dann wird das kleine Feld static_info_country der fe_users Tabelle anstelle des country Feldes verwendet. Das ist notwendig, wenn Sie eine Select Box zur Auswahl des Landes machen.

Produkt Eigenschaften:

Die Produkteigenschaften werden im Systemfolder für die jeweiligen Artikel im Backend eingegeben.



Farbe, Größe, Zusätzliches und Staffelung

Damit Sie Variationen von Produkten eingeben können, müssen Sie die Werte durch Strichpunkt getrennt eingeben. So müssen Sie für T-Shirts mit verschiedenen Farben und Größen "rot;grün;blau" als Variante 1 und "S;X;XL;XXL" als Variante 2 eingeben. Farbe (Variante 1) und Größe (Variante 2): geben Sie hier Werte mit den Titeln ein, die durch Strichpunkte 'colon ';' getrennt werden wie z.B.

```
M;L;XL;XXL
```

Ändern Sie das Template, damit das unterstützt wird:

```
<!-- ###display_variant1### -->
###GW2B### Color: ###GW2E### <SELECT style="font-size: 10px"
name="###FIELD_COLOR_NAME###" rows="1">###PRODUCT_COLOR###</SELECT><br>
<!-- ###display_variant1### -->
<!-- ###display_variant2### -->
###GW2B### Size: ###GW2E### <SELECT style="font-size: 10px"
name="###FIELD_SIZE_NAME###" rows="1">###PRODUCT_SIZE###</SELECT><br> <!-- ###display_variant2### -->
```

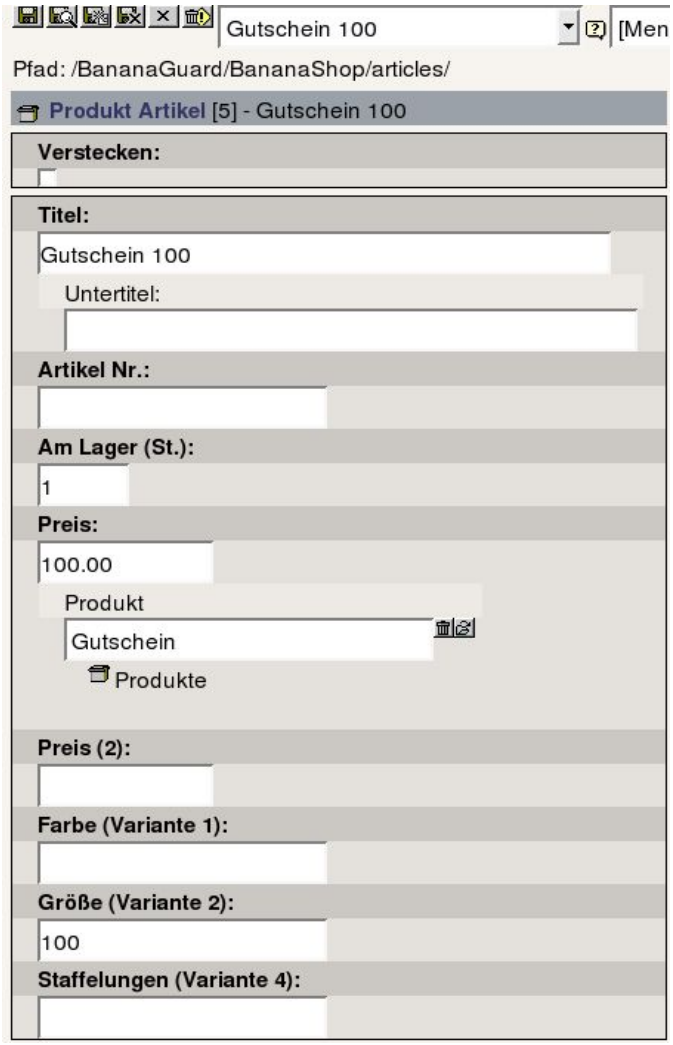
Wenn Sie die Farben und Größen nicht auswählbar machen wollen, dann müssen Sie im Template Setup selectColor oder

selectSize auf 0 setzen und im Template selbst die entsprechende ###display_variant### entfernen. Sie können diese Marker jedoch zur Darstellung der Farben verwenden, auch wenn keine unterschiedlichen Farben gewählt werden können. (dann nur die Markierungen ###PRODUCT_COLOR### und ###PRODUCT_SIZE### verwenden). Denn im Setup wurden ja Farben und Größen auf 0 gesetzt.

Der Feldname aller Felder in der Listenansicht und der Warenkorbansicht wird mit dem Marker ###FIELD_NAME### angesprochen. Dieser Marker wird intern im Shop durch einen Feldnamen ersetzt, damit das Fels ausgelesen und richtig zugeordnet werden kann.

Wenn Sie Produkte mit verschiedenen Farben, Größen und Staffelungen nach Mengen haben, dann müssen Sie im Bereich BASKET_TEMPLATE den Marker ###FIELD_NAME### durch den Marker ###FIELD_NAME_BASKET### ersetzen.

In der Listenansicht des Webmoduls können Sie nun die bereits vorhandenen Artikel um die Varianten, z.B. Farbe, ergänzen. Sie wählen dazu die Tabelle Produkt Artikel aus.



Sie müssen im Template Setup useArticles=1 setzen, um die Artikeleigenschaften zu verwenden.

Wenn Sie bereits Produkte angelegt haben, diesen bereits in Produkteigenschaften verschiedene Farben zugeordnet haben, verfügen Sie jedoch nur über eine Artikelnummer und einen Preis etc. zu diesem Produkt. Sie benötigen jedoch für Produkte z. B. unterschiedlicher Farbe auch unterschiedliche Artikelnummern und vielleicht Preise etc. Diese Zurodnungen erfolgen in den Artikeleigenschaften (Webmodul, Liste, Produkte Artikel – Tabelle). Der Vorteil ist, dass Sie die “neuen” Produkte nicht neu anlegen müssen, sondern die Eigenschaften vererbt werden. Sie geben also nur noch die Farbe, neue Artikelnummer und evtl. neuen Preis ein. Alle anderen Angaben zum Produkt werden übernommen. Lassen Sie ein Feld leer, so wird der Feldinhalt aus den Produkte Eigenschaften übernommen.

Es macht einen Unterschied, ob ein Artikel mit verschiedenen Farben oder Größen in der Produktliste oder im Warenkorb angezeigt wird. Normalerweise wird die Farbe und Größe eine Produktes über eine Auswahlbox ausgewählt. In der Produktliste wird die Anzahl für die erste Farbe/Größe angezeigt. Im Warenkorb jedoch werden die Produkte in allen Varianten einzeln aufgelistet. Die Listes der Produkte im Warenkorb und in der Auswahlseite sind also beinahe gleich, aber sie werden anders gehandhabt. Sie müssen selectSize und selectColor deaktivieren, wenn Sie keine Auswahlboxen in der Produktliste haben wollen.

Gewicht, Sperrgut und Spezialanfertigung

Jedes Produkt kann ein Gewicht haben und als Sperrgut deklariert werden (Ja/Nein). Das Gesamtgewicht der bestellten Artikel wird berechnet und kann in der Preisberechnung für die Versandkosten berücksichtigt werden.

Wenn ein Artikel als Sperrgut markiert worden ist, dann wird bei dessen Anzeige eine Warnung als 'Sperrgut' angezeigt. ('bulkilyWarning' im Setup). Durch das Setzen von 'bulkilyAddition' im Setup können Sie zusätzliche Versandkosten für dieses Sperrgut dazuaddieren. Ein Spezialanfertigung beinhaltet keine weitere Funktionalität. Es wird nur eine Markierung (ähnlich wie bei BulkilyWarning) geschrieben. Sie können hier einen Link auf die Email Seite setzen.

Beispiel:

```
specialPreparation = special preparation is possible! <A
HREF="?id=999&tt_products=###PRODUCT_ID###">Order here</A>.
```

Einfache Lagerverwaltung

Das Feld inStock kann für eine Lagerverwaltung verwendet werden. Wenn "in stock" auf "0" gesetzt ist, wird der Artikel für die Benutzer nicht mehr sichtbar sein. Nach jeder Bestellung wird die Anzahl der bestellten Artikel von "in stock" abgezogen.

Wenn ein neuer Artikel angelegt wird, wird dessen Wert standardmäßig auf '1' gesetzt. Wenn Sie 'alwaysInStock=1' gesetzt haben, dann werden diese Artikel immer verfügbar und sichtbar sein. Ansonsten wird die Anzahl der Artikel bei jeder Bestellung reduziert, auch auf negative Werte. Ein negativer Wert bedeutet, daß ein Produkt nun schleunigst beim Lieferanten bestellt werden muß.

Sie können, um einzelne Artikel immer auf Lager zu halten, die entsprechende Checkbox ankreuzen. Dies gilt aber nur für den jeweiligen Artikel. Sie können die Einheiten der Waren auf dem Lager über "inStockPieces" setzen, wie z.B. "Stück". oder Bundle. Diese Mengeneinheit kann jedoch nur einmal gesetzt werden und gilt dann für alle Produkte.

Mehrere Steuersätze

Zu jedem Artikel können Sie den Steuersatz eingeben.

Konfiguration

FAQ

- Fragen zur Konfiguration - ToDo

Dateien

Datei:	Beschreibung:
class.tx_ttproducts.php	Haupklasse zur Darstellung der Produktliste oder des Warenkorbs. Rufen Sie es von einem USER cObject über 'userFunc = user_products->main_products' auf.
products_comp_calcScript.inc	Beispiel Berechnungsskript
products_template.tmpl products_template_htmlmail.tmpl	Beispiel Templates auf Englisch. '..._htmlmail.tmpl' ist ein HTML-wrap für the versendeten HTML-Emails
products_template_dk.tmpl	Beispiel Template auf Dänisch
example_template_bill_de.tmpl	bananaguard.de Template mit Beispielen für Rechnung und Lieferschein auf Deutsch
products_template_fi.tmpl	Beispiel Template auf Finnisch
products_template_fr.tmpl	Beispiel Template auf Französisch
products_template_se.tmpl	Beispiel Template auf Schwedisch
product_detail.tmpl product_proefpakketten.tmpl shop-a-box.tmpl producten.tmpl	Beispiel Templates für Gutscheinpunkte-System und Freundschaftswerbung auf Holländisch
products_css_en.html	CSS basierte Template. Verwenden Sie das, wenn Sie ein barrierefreies Shop erstellen wollen.
'Bearbeitungsskripts' für den Zugriff auf externe Bezahl-Gateways	
payment_DIBS.php	Skript für eine Schnittstelle zu DIBS (Danish Internet Payment System) in Dänemark. Sie könne Sie unter http://www.architrade.com/uk/ erreichen.
payment_DIBS_template.tmpl	Template Datei für das DIBS Bezahlsystem.

[tsref:(cObject).TEST]

Beschreibung

Eingebauter Warenkorb und Produktanzeige unter TYPO3. Es beinhaltet eine freie Schnittstelle, mit der Sie Zugriff auf Ihre eigene Implementierung eines Bezahlsystems haben können.

Im Moment gibt es eine Implementierung für DIBS in Dänemark, unter www.architrade.dk.

Referenz

class.tx_ttproducts.php properties

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
templateFile	resource	Die Template Datei. Siehe Beispiel in 'media/scripts/products_template.tmpl'	
pid_list	string /stdWrap	Die pids, aus welchen die Kategorien, Produkte und so weiter gelesen werden. Standard ist die aktuelle Seite. Hier können mehrere pids durch Beistrich getrennt eingegeben werden.	
defaultCode	string	Der Standard Code, wenn keiner für das Plugn eingetragen ist (siehe unten). Normalerweise wird das nicht gesetzt und ein Hilfe-Bildschirm wird angezeigt. **Beispiel:** `plugin.tt_products.defaultCode = HELP`	

<table>
<tr><th>Eigenschaft:</th><th>Datentyp:</th><th>Beschreibung:</th><th>Voreinstellung:</th></tr>
<tr><td>code</td><td>string /stdWrap</td><td>Code zur Bestimmung der Darstellung und Aktion, die im Shop abläuft. Anstelle dieser Codes (Großbuchstaben) werden die Einstellungen im Shop-Plugin über Flexforms durchgeführt.

<u>Produkte:</u>
<table><tr><td>Liste (LIST)</td><td>Auflisten der Produkte</td></tr><tr><td>Liste Geschenke (LISTGIFTS)</td><td>Auflisten der Geschenke</td></tr><tr><td>Liste Besonderheiten (LISTHIGHLIGHTS)</td><td>Auflisten der Produkte, die besonders attraktiv sind</td></tr><tr><td>Liste Aktionen (LISTOFFERS)</td><td>Auflisten der Produkte, die als Angebote gelten</td></tr><tr><td>Liste Neue Artikel (LISTNEWITEMS)</td><td>Auflisten der zuletzt eingetragenen Produkte</td></tr><tr><td>Einzelansicht (SINGLE)</td><td>Einzelansicht eines Artikels (auch in Liste möglich) oder GET/POST 'tt_products'</td></tr><tr><td>Suche (SEARCH)</td><td>Suchdialog zur Suche von Produkten</td></tr></table>
<u>Währung:</u>
<table><tr><td>Selector - CURRENCY currency selector</td><td>Auswahlbox für Währungen</td></tr></table>
<u>Warenkorb:</u>
<table><tr><td>Inhalt (BASKET)</td><td>zeigt den Warenkorb mit den Artikeln an, die gekauft werden sollen.
Der 'BASKET' Code hat verschiedene Funktionen. Aber wenn man die spezifischen Codes INFO, PAYMENT und FINALIZE verwendet, dann kann diese Funktionalität auf mehrere Seiten aufgespalten werden</td></tr><tr><td>Mini (OVERVIEW)</td><td>Ein Mini-Warenkorb, der nur die Anzahl der Produkte enthält</td></tr><tr><td>Eingabe Kundendaten (INFO)</td><td>Eingabe der Adreßinformation</td></tr><tr><td>Eingabe Bezahlung (PAYMENT)</td><td>Überprüfung der Bestelldaten, Bezahlung über Bezahlsystem</td></tr><tr><td>Bestellung abschließen (FINALIZE)</td><td>schließt die Bestellung ab und versendet Emails Information über die Verwendung von tt_products - Seite mit Dank für die Bestellung</td></tr></table>
<u>Bestellung:</u>
<table><tr><td>Verwaltung (TRACKING)</td><td>Nachverfolgen der Bestellung, der Rechnung und des Lieferscheins</td></tr><tr><td>Rechnung (BILL)</td><td>erzeugt eine Rechnungs-Datei</td></tr><tr><td>Lieferschein (DELIVERY)</td><td>erzeugt eine Lieferschein-Datei</td></tr><tr><td>Liste (ORDER)</td><td>Anzeige der Bestellungen für Frontend Benutzer</td></tr></table>
<u>Allgemein:</u>
<table><tr><td>Memo (MEMO)</td><td>Erinnerung an Produkte für Frontend Benutzer</td></tr><tr><td>Hilfe (HELP)</td><td>Kurzanleitung</td></tr></table>
<u>Kategorien:</u>
<table><tr><td>Liste (LISTCAT)</td><td>Liste der Kategorien</td></tr></table>
<u>Artikel:</u>
<table><tr><td>Liste (LISTARTICLES)</td><td>Liste der Artikel</td></tr></table>
Verwenden Sie die CODE-Felder nur für TypoScript Setups.</td><td>HELP</td></tr>
<tr><td>defaultProductID</td><td>int+</td><td>Die standardmäßige Produkt Id für die Einzelansicht wird angezeigt, wenn der Link auf das Skript keinen Parameter 'tt_products' gehabt hat.
Setzen Sie diesen Standard-Wert, wenn Sie eine Fehlermeldung wie “Falscher Parameter, GET/POST var 'tx_ttproducts_pi1[product]' wurde nicht angegeben” erhalten.</td><td></td></tr>
<tr><td>defaultCategoryID</td><td>int+</td><td>Die standardmäßige Kategorie Id für die Listenansicht wird verwendet, wenn der Link auf das Skript keinen Parameter 'tx_ttproducts_pi1[cat]' gehabt hat. Verwenden Sie das, wenn Sie standardmäßig nur Produkte dieser Kategorie in der Listenansicht angezeigt haben wollen.</td><td></td></tr>
</table>

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
recursive	int+	Anzahl der rekursiven Ebenen für die Listenanzeige von Produkten	
domain	string	Die Url des Geschäfts. Wenn sie nicht gesetzt ist, wird sie automatisch erkannt. Damit werden die ###DOMAIN### Markierungen ersetzt.	

altMainMarkers	*(array of strings)*	Spezifizierung von alternativen Unter-Markierungen. Für die verschiedenen Haupt Template Designs des Shop Systems.
		Diese Liste an Unter-Markierungen können Sie überschreiben:
		
		`TRACKING_WRONG_NUMBER`
		`TRACKING_ENTER_NUMBER`
		`BASKET_REQUIRED_INFO_MISSING`
		`BASKET_TEMP`
		`ITEM_SINGLE_DISPLAY_RECORDINSERT`
		`ITEM_SINGLE_DISPLAY`
		`ITEM_SEARCH`
		`ITEM_LIST_TEMPLATE`
		`ITEM_LIST_GIFTS_TEMPLATE`
		`ITEM_SEARCH_EMPTY`
		`BASKET_TEMPLATE`
		`BASKET_INFO_TEMPLATE`
		`BASKET_PAYMENT_TEMPLATE`
		`BASKET_ORDERCONFIRMATION_TEMPLATE`
		`EMAIL_PLAINTEXT_TEMPLATE`
		`BILL_TEMPLATE`
		`DELIVERY_TEMPLATE`
		
		`/+ stdWrap`
		
		Beispiel:
		`altMainMarkers.BASKET_TEMPLATE = BASKET_DESIGN2`
		`altMainMarkers.BASKET_TEMPLATE.wrap = ### | ###`
		
		Dieses Beispiel ändert die Haupt Markierungen der regulären Warenkorb Anzeige vom standardmäßigen ###BASKET_TEMPLATE### in das benutzerdefinierte Design ###BASKET_DESIGN2### (in derselben HTML-Template Datei)	

stdSearchFieldExt	*list of fields*	Standard Suchfelder Die standardmäßige interne Liste ist title,subtitle,note. Hier können Sie Ihre eigenen Suchfelder angeben.	
limit	int+	Max. Anzahl an angezeigten Artikeln Nicht mehr Artikel werden pro Seite angezeigt.	50
limitImage	int+	Max. Anzahl an Bildern. Die maximale Anzahl an Bildern die für einen Artikel in der Listenansicht angezeigt wird.	1
limitImageSingle	int+	Die maximale Anzahl an Bildern die für einen Artikel in der Einzelansicht angezeigt wird.	1

~~usePageContentImage~~	~~boolean~~	**Veraltet.** Siehe Konfiguration der Artikel/Produkte
		Verwenden Sie stattdessen
		`plugin.tt_products.conf.tt_products.ALL.fetchImage {`
		`  type = foreigntable`
		`  table = tt_content`
		`}`	

separateImage	boolean	Normalerweise werden alle Bilder zusammen angezeigt. Mit separateImage=on kann eine Markierung ###PRODUCT_IMAGE3### für jedes einzelne Bild angegeben werden. Die Zählung beginnt bei 1.	
image	IMAGE cObject	Die Bild Konfiguration für die Einzelansicht	

listImage	IMAGE cObject	Die Bild Konfiguration für die Listenansicht
		`listImage >`
		`listImage {`
		`  altImgResource.import = uploads/media/`
		`  altImgResource.import.field = media`
		`  altImgResource.import.listNum = 0`
		`  altText.data = field:title`
		`}`
		
		Auf diese Art werden die Bilder nicht in das Verzeichnis */typo3temp/* kopiert und von dort angezeigt. (Das bereitet Schwierigkeiten mit transparenten Hintergründen), sondern direkt von /uploads/pics/.
		Die Zeile mit 'altText' bewirkt, daß der Titel als alternativer Text angezeigt wird.	

basketImage	IMAGE cObject	Die Bild Konfiguration für den Warenkorb	
datasheetIcon	IMAGE cObject	Das Symbolbild für das Datenblatt. ###ICON_DATASHEET### wird dadurch ersetzt.	
basketPic	string	URL Link auf das Warenkorb Bild	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
clickIntoBasket	boolean	Falls gesetzt, wird sofort in die Warenkorbansicht gewechselt, sobald ein Produkt in den Warenkorb gegeben worden ist. Das funktioniert nur, wenn PIDbasket gesetzt worden ist.	
clickIntoSubmenu	boolean	Falls gesetzt, werden die Untermenüeinträge der LISTCAT Kategorieliste nur für die aktuelle Kategorie angezeigt.	
basketMaxQuantity	int+ / string	Die maximale mögliche Anzahl eines Produkts, die in den Warenkorb gelegt werden kann. 'inStock': Nur maximal soviele Produkte können in den Warenkorb gelegt werden als sich momentan auf Lager befinden.	100000
quantityIsFloat	boolean	Wenn gesetzt, ist die Anzahl eine Fließkommazahl.	
noImageAvailable	resource	Die Bilddatei, die angezeigt wird, wenn ein Produkt kein Bild hat. Dieses Bild wird über das IMAGE cObject generiert, das im aktuellen Bild des Produktes angezeigt wird. Es ist eines der oberen IMAGE cObject.	
displayListCatHeader	boolean	Anzeige der Kategorie Überschrift in Listen Wenn diese Option gesetzt ist, dann werden die Kategorie Überschriften (Seitentitel) automatisch in den Produktlisten angezeigt. Das ist manchmal nicht erwünscht, wenn Sie die Überschriften selber in den Inhaltselementen von "list" ausgeben.	1
displayBasketCatHeader	boolean	Anzeige der Kategorieüberschrift im Warenkorb Wenn diese Option gesetzt ist, dann werden die Kategorieüberschriften (Seitentitel) automatisch in den Listen im Warenkorb angezeigt.	0
displayCatListType	string	Bestimmt welches HTML Tag für die Anzeige der Kategorien in der Liste verwendet wird. mögliche Werte: ❿ ul ❿ select ❿ null	ul
displayBasketColumns	int+	Anzahl an Spalten in den Ansichten LIST, SEARCH. Es wird eine HTML Tabelle aufgebaut. Dazu müssen spezielle Template Markierungen verwendet werden. Die ITEM_SINGLE_PRE_HTML und ITEM_SINGLE_POST_HTML müssen vor und nach den <TD>-</TD> Tags stehen, damit die Tabelle richtig erzeugt wird. **Beispiel:** `<!-- ###ITEM_SINGLE### begin-->` `###ITEM_SINGLE_PRE_HTML###` `<TD height="150" bgcolor="###GC1###" valign="bottom" align="center">` `<!--###LINK_ITEM###-->` `###GW1B###<b>###PRODUCT_TITLE###</b>###GW1E###` `<!--###LINK_ITEM###-->` ` ` `###PRODUCT_IMAGE###  ` `###GW1B### item count: ###GW1E### <INPUT size="3" maxlength="4" type="text" name="###FIELD_NAME###" value="###FIELD_QTY###">` `<p> </TD>` `###ITEM_SINGLE_POST_HTML###` `<!-- ###ITEM_SINGLE### end -->`	
CSS	*siehe unten*	CSS Einstellungen	
conf	*siehe unten*	Konfiguration der Tabellen	
NoSingleViewOnList	boolean	Normalerweise wird der Link auf die Einzelansicht in der Seite mit dem LIST Code generiert. Wenn Sie aber Ihre eigenen Seiten für die Einzelansicht mit dem SINGLE Code erzeugen wollen, dann müssen Sie das auf 1 setzen.	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

itemMarkerArrayFunc	function-name	Jedes Mal, wenn ein Produkt angezeigt wird, sei es im Warenkorb oder in der Einzelansicht, wird die Methode getItemMarkerArray() in user_products aufgerufen. Diese Funktion füllt ein Array aus und gibt es zurück, das sogenannte markerArray(), mit Schlüsseln/Werten für die Template Ersetzung.
		Wenn Sie hier einen gültigen Funktionsnamen eingeben (siehe Datentyp 'function-name' für Details), dann wird dieses Array anschließend als zweiter Parameter an diese Funktion übergeben. Der erste Parameter für die itemMarkerArrayFunc ist die TypoScript Eigenschaft.
		
		Parent PHP-Object Referenz:
		.parentObj property ist eine hardcodierte Referenz zum aufrufenden user_products Objekt (PHP).
		
		Beispiel:
		(vorausgesetzt, daß eine Funktione oder Klasse inkludiert ist!)
		
		`itemMarkerArrayFunc = user_addFieldsMarkerArr`
		`itemMarkerArrayFunc.simpleOption = 1`	

PIDitemDisplay	int+/*Array of integers*	PID für die EInzelansicht.
		Wenn Sie wollen, daß eine eigene Seite für die Einzelansicht von Produkten zuständig sein soll, dann geben Sie hier die PID (page-uid) an. Sie können hier Bedingungen in mehreren Zeilen angeben. Der where String kann AND Verknüpfungen enthalten. Die pid der ersten erfüllten Bedingung wird zurückgegeben.
		
		`PIDitemDisplay {`
		`10.type = sql`
		`10.where = color=red`
		`10.pid = 142`
		`20.type = sql`
		`20.where = color=white`
		`20.pid = 143`
		`}`
		
		Wenn Sie den Typ auf pid setzen, dann wird die pid des Produkt Datensatzes verwendet.
		`PIDitemDisplay {`
		`10.type = pid`
		`}`	

PIDlistDisplay	int+/*Array of integers*	PID für die Listenanzeige der Produkte Ähnlich zu PIDitemDisplay, aber hier wird die Kategorietabelle verwendet.	
PIDsearch	int+	PID für die Suchenseite Wenn Sie möchten, daß alle Produktsuchen auf eine spezielle Seite gelangen, dann tragen Sie hier die PID ein! ACHTUNG: Wenn Sie diese PID setzen, müssen alle Seiten mit Suchabfragen ein Plugin mit dem Code "SEARCH" haben.	
PIDbasket	int+	PID für die Warenkorb Seite. Wenn Sie wollen, daß nach der Änderung der Anzahl zu einem Artikel eine Seite aufgerufen wird, dann geben Sie das hier an.	
PIDstoreRoot	int+	PID für die Wurzel des Shops. Das ist die PID der rootPage des Shops. Wenn diese nicht gesetzt ist, wird das Shop über alle Seiten ausgehend von der Root der Homepage arbeiten. Aus Performance-Gründen sollte das gesetzt sein, wenn mit Seiten als Kategorien gearbeitet wird.	
PID_sys_products_orders	int+	PID für die Datensätze in sys_products_orders. Standardmäßig erhalten sie die pid der payment (finalize) Seite.	
PIDGiftsTable	int+	PID für die tt_products_gifts Tabelle. Die Geschenk Bestellungen werden hier gespeichert.	
PIDinfo	int+	PID für die info Seite, bei welcher Name und Adresse eingegeben werden	
PIDfinalize	int+	PID für die Seite nach der Bestätigungsseite, wo der Benutzer seine Daten überprüft und den AGB zugestimmt hat	
PIDthanks	int+	PID für die Dankesseite. BASKET_ORDERTHANKS_TEMPLATE wird hier verwendet. Sonst ist es ident zu PIDfinalize. Daher darf PIDfinalize nicht gleichzeitig gesetzt sein!	
PIDtracking	int+	PID für die Nachverfolgung einer Bestellung	
PIDbilling	int+	PID für die Erzeugung der Rechnung	
PIDdelivery	int+	PID für die Erzeugung des Lieferscheins	
PIDmemo	int+	Die ID der Memo Seite	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
PIDagb	int+	Die PID einer Seite mit den Allgemeinen Geschäftsbedingungen (“AGB”) Nur wenn hier eine Seiten-Id eingetragen ist, wird geprüft, ob die AGB vom Benutzer durch das Setzen eines Häkchen vor 'Akzeptieren der AGB' akzeptiert worden sind.	0
PIDuserFolder	int+	Der sysfolder, wo neue Benutzer abgespeichert werden sollen.	116
paymentActivity	string	Zeitpunkt im Shop, wann die Bezahlung über ein eigenes Bezahlskript durchgeführt werden soll. Mögliche Werte: payment, finalize	finalize

advanceOrderNumberWithInteger	int+ , int+	Wenn dieser Wert gesetzt ist, dann wird beim Erzeugen einer jeden Bestellung der Zähler der Bestellungen um eine Zufallszahl zwischen [erste-Integer] und [zweite Integer] erhöht, um etwas zu mogeln.
		
		Beispiel:
		`1,10 (Das wird den Zähler um einen Zufallswert zwischen 1 und 10 erhöhen)`
		`5,5 (Das wird den Zähler jedes Mal um 5 erhöhen)`	

alwaysAdvanceOrderNumber	boolean	Wenn das gesetzt ist, wird die Bestellnummer immer erhöht, und die leeren Bestellnummern werden nicht mehr wiederverwendet. Sie müssen das verwenden, wenn Sie ein Bezahlungsskript verwenden, das keine doppelten Bestellnummern akzeptiert.	

parseFunc	->parseFunc
	
	mit 'CSS styled content' nicht verwendet
		Wenn die Extension 'CSS styled content' installiert ist, dann müssen Sie die Einstellung in lib.parseFunc_RTE machen statt hier.
		Die Produktbeschreibungen werden mit dieser Funktion überprüft. Wenn Sie es z.B. zulassen wollen, daß im Feld 'Bemerkung' auch HTML-Tags zur Erzeugung einer Tabelle vorkommen dürfen, dann müssen Sie die parseFunc.allowTags oder auch die denyTags entsprechend setzen.
		Damit die unter RTE eingegebenen HTML-Anweisungen richtig angezeigt werden, müssen Sie Folgendes im Root Template eingeben.
		
		Beispiel:
		`keepNonMatchedTags = 1`
		`RTE.default.proc.preserveTables = 1`
		
		Beispiel:
		`parseFunc.allowTags = table,tr,td,b,i,u,a,img,br,div,center,pre,font,hr,sub, sup,p,strong,em,li,ul,ol,blockquote,strike,span,h1,h2, h3,h4,h5,h6`
		`parseFunc.denyTags = *`
			styles.content.parseFunc

categoryHeader	cObject	Erzeugt die Überschrift zu den Kategorien.
		
		Beispiel:
		
		`categoryHeader = TEXT`
		`categoryHeader.current = 1`	

TAXpercentage	double	Prozente des Steuersatzes.
		Double Wert (!) (das bedeutet, Sie müssen den Punkt '.' statt dem Komma verwenden)
		
		Beispiel:
		`# Der deutsche Steuersatz beträgt TAX 16%:`
		`TAXpercentage = 16.00`	

TAXincluded	boolean	Setzen Sie das, wenn die Steuer in den Preisen in der Datenbank miteingerechnet ist! (... und natürlich: Löschen Sie das, wenn die Steuer nicht in den Preisen in der Datenbank enthalten ist, sondern bei der Anzeige der Preise noch dazugerechnet werden muß.) Alle Abläufe im Shop berücksichtigen diese Einstellung bei den Preisberechnungen.	1
priceDec	int+	Anzahl der Nachkommastellen im Preis	
priceDecPoint	string	Dezimalpunkt im Preis	
priceThousandPoint	string	Trennzeichen für die Tausender, falls gewünscht	

priceNoReseller	int+	Preis Nummer für Wiederverkäufer. Im Moment kann hier nur 2 eingegeben werden. price2 wird aber nur dann verwendet, wenn der Wert größer als 0 ist.
		
		Beispiel:
		`[usergroup = 1]`
		`priceNoReseller = 2`
		`[global]`
		`...`	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

currencySymbol	string	Währungseinheit. Es wird in der Administration verwendet.
		
		Beispiel:
		`EUR`
		`DKR`
		`USD`
		`$`	

lockLoginUserInfo	boolean	Wenn das gesetzt ist, und ein Benutzer sich eingeloggt hat, dann wird die Adreßinformation des Benutzers für die Rechnungsanschrift verwendet.	
loginUserInfoAddress	boolean	Wenn lockLoginUserInfo gesetzt ist, dann bewirkt dieser Schalter, daß das Adreßfeld aus den Feldern address, country, zip und city aus fe_users	
requiredInfoFields	*list of string*	Liste der Felder, die für die Adreßinformation benötigt werden. Dieses Beispiel beschreibt alle Möglichkeiten. **Beispiel:** `requiredInfoFields = name,address,telephone,fax,email,company,city,zip,state,country`	
~~orderBy~~	~~string~~	siehe Artikel/Produkt Konfiguration **Veraltet.** Verwenden Sie stattdessen `plugin.tt_products.conf.tt_products.ALL.orderBy = title`	
~~orderByCategoryTitle~~		**Veraltet.** Verwenden Sie stattdessen `plugin.tt_products.conf.tt_products_cat.ALL.orderBy = title`	
orderByItemNumberSg	boolean	Wenn die Einzelansicht nach der Artikelnummer ItemNumber anstelle der uid sortiert werden soll; wird für ###LINK_NEXT_SINGLE### und ###LINK_PREV_SINGLE### verwendet	
orderNumberPrefix	string	Präfix für die Bestellnummern. Max 10 Zeichen. Wenn dieser Text mit '%' beginnt, dann wird der Rest im PHP Datumsformat angegeben, um ein Zeit/Datum zu ergeben.	
orderEmail_from	string	Absender Email Adresse für Bestätigungsemail an den Kunden	
orderEmail_fromName	string	Absender Name für Bestätigungsemail an den Kunden	
orderEmail_to	*list of email-addresses*	Eine durch Beistrich getrennte Liste der Empfänger der Bestellungsemail Shop und Administrator/Supervisor Email Adressen müssen hier eingetragen werden!	
orderEmail_toDelivery	boolean	Falls gesetzt, wird die Bestätigungsemail an die Zustell-Email-Adresse und nicht an die Rechnungs-Email-Adresse gesendet.	
orderEmail_subject	string	Inhalt der Betreff-Zeile, falls die erste Zeile in ###EMAIL_PLAINTEXT_TEMPLATE### leer ist.	
orderEmail_htmlmail	boolean / string	Falls gesetzt, wird die Bestell Email im HTML Format gesendet. Wenn orderEmail_htmlmail.removeImagesWithPrefix gesetzt ist, dann werden die Bilder samt ihren HTML Tags nicht in der Email gesendet.	
email_notify_default	boolean	Die Benachrichtungsemail an den Kunden ist standardmäßig aktiviert. Er kann das selber später im Modul Bestellungsverwaltung ändern.	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

statusCodes	*array of integers*	Zustandscodes für das Modul Bestellungsverwaltung.
		Nummern über 100 löschen eine Bestellung aus der Bestellungsverwaltung.
		Nummer 0 ist der Zustand einer nicht abgeschlossenen Bestellung (und nicht abgeschlossene Bestellungen in der Datenbank können später als Müll angesehen werden...)
		Nummern zwischen 50-59 sind zur Auswahl für den Kunden.
		Nummern zwischen 60-69 sind Spezialfunktionen wie Geschenke für den Kunden.
		Nummer 1 wird dafür verwendet, wenn eine Bestellung bestägit worden ist, und deshalb von 0 auf 1 wechselt. Nummer 1 kann vom Shop Admin nicht ausgewählt werden.
		Das wird in die Markierungen ###STATUS_OPTIONS### geschrieben.
		
		Beispiel:
		`statusCodes {`
		`  1 = Bestellung vom Kunden bestätigt`
		`  2 = Bestellung ist vom Shop erhalten und akzeptiert worden`
		`  10 = Das Shop wartet auf Waren von einem Lieferanten`
		`  11 = Das Shop wartet auf die Bezahlung durch den Kunden`
		`  12 = Das Shop wartet auf Waren vom Kunden`
		`  13 = Die Bestellung wurde bezahlt`
		`  20 = Die Waren wurden an den Kunden versendet`
		`  21 = Geschenk Gutscheine wurden an den Kunden gesendet`
		`  30 = Andere Nachricht vom Shop`
		`  50 = Der Kunde bittet um die Löschung der Bestellung`
		`  51 = Meldung vom Kunden an das Shop`
		`  60 = Senden der Geschenkgutscheine an die Empfänger`
		`  100 = Bestellung versendet und geschlossen`
		`  101 = Bestellung geschlossen`
		`  200 = Bestellung gelöscht`
		`}`	

update_code	string	Das 'Paßwort', das vom Shop Administrator verwendet wird, um in die Bestellungsverwaltung des Shops im Frontend zu gelangen. Das Paßwort Formularfeld wird angezeigt, wenn ein Backend-Benutzer eingeloggt ist. Trotzdem wird hierfür nochmals ein Paßwort verlangt.	password
statusDate_stdWrap	->stdWrap	stdWrap für das Datum des Zustands **Beispiel:** `statusDate_stdWrap.strftime = %d-%m-%y %H:%M`	
orderDate_stdWrap	->stdWrap	stdWrap das Datum der Bestellung **Beispiel:** `orderDate_stdWrap.strftime = %d-%m-%y`	
displayCurrentRecord	boolean	Wenn das gesetzt ist, werden verschiedene Einstellungen in der Anzeige der Bestellung verändert. Die Anzeige wird aus einem $cObj->data erzeugt. Wenn das gesetzt ist, wird die Untermarkierung ###ITEM_SINGLE_DISPLAY_RECORDINSERT### anstelle der gewöhnlichen ###ITEM_SINGLE_DISPLAY### verwendet.	
externalProcessing	cObject	Dieses cObject kann zum Aufruf einer Funktion verwendet werden, die den Warenkorb verändert. Diese Verarbeitung kann in Abhängigkeit von den Einstellungen in einem anderen externen Shop-System erfolgen. Die Ausgabe wird in an den Anfang der Ausgabe der Warenkorbseite angehängt. Dieses cObject führt jedes mal die Methode main_products method in der Klasse user_products in class.tx_ttproducts.php aus. Sie wird immer vor jeder anderen Verarbeitung im Shop aufgerufen. Siehe Klassenbeschreibung im Code.	
externalProcessing_final	cObject	cObject für das Template der Bestellbestätigung	
externalFinalizing	cObject	Dieses cObject kann zum Aufruf einer Funktion verwendet werden, die Einstellungen in einem externen Bestellsystem vornimmt. Dies ist eine Schwester der oben angegebenen Funktionen. Sie sollten irgendwie gemeinsam verwendet werden. Diese Funktion wird unmittelbar nach dem Aufruf der finalize-Funktion aufgerufen. Diese Funktion ist zum Beispeil dafür geeignet, um jeden externen Warenkob, der über das .externalProcessing cObject möglich ist, löschen zu können. **Achtung:** Die Ausgabe wird nirgends angezeigt.	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
wrap1	-> stdWrap	Globaler Wrap 1. Das wird auf die Markierungen ###GW1B### und ###GW1E### verteilt. Geben Sie hier keinen normalen Wert ein, sondern verwenden Sie den Trennstrich zur Kennzeichnung der beiden Teile. **Beispiel:** `wrap1.wrap = <B> |</B>`	
wrap2	-> stdWrap	Globaler Wrap 2 (siehe oben) Markierungen ###GW2B### und ###GW2E###	
selectColor	boolean	Wenn gesetzt, dann ist die Farbe eines Produkt über eine Auswahlbox wählbar.	1
selectSize	boolean	Wenn gesetzt, dann ist die Größe eines Produkt über eine Auswahlbox wählbar.	1
selectDescription	boolean	Wenn gesetzt, dann ist die Beschreibung eines Produkt über eine Auswahlbox wählbar.	1
selectGradings	boolean	Wenn gesetzt, dann ist die Staffelung eines Produkt über eine Auswahlbox wählbar.	1
color1	string /stdWrap	Wert für ###GC1### Markierungen (Globale Farbe 1)	
color2	string /stdWrap	Wert für ###GC2### Markierungen (Globale Farbe 2)	
color3	string / stdWrap	Wert für ###GC3### Markierungen (Globale Farbe 3)	
nl2brNote	boolean	Wenn gesetzt, werden im Feld 'Bemerkung' (note) Zeilenumbrüche durch das HTML-Tag ersetzt.	1
useArticles	integer	❶ 0: Wenn Sie keine Einträge in der Artikel Tabelle machen wollen, dann muß das auf 0 gesetzt beleiben. ❶ 1: Wenn Sie Produkte in verschiedenen Variationen verwenden wollen und dafür eigene Preise in der Artikel Tabelle angelegt haben, dann müssen Sie das auf 1 setzen. Das Feld instock der Artikel Tabelle wird anstelle jenes der Produkt Tabelle verwendet. ❶ 2: Wie 1 aber es muß zwischen einem Produkt und seinen Artikeln keine übereinstimmenden Felder bei den Varianten geben.	
priceTagObj	IMAGE cObject	Bild oder cObject für die Anzeige des Preises	
usePriceTag	boolean	Wenn gesetzt, wird das priceTagObj verwendet. Wenn nicht, dann wird der Preis als normaler Text ausgegeben.	0
useStaticInfoCountry	boolean	Wenn gesetzt, dann wird das kurze fe_user static_info_country Feld als Land anstelle des country Feldes verwendet.	0
whereGift	string	Diesee SQL Where Bedingung bestimmt, welche Produkte als Geschenke (Gutscheine) anzusehen sind.	
max_note_length	int+	Die Bemerkung zu einem Artikel wir in der Listendarstellung nach max_note_length Zeichen abgebrochen. Dahinter werden noch 3 Punkte angehängt... Das kann aber unschön sein, wenn Sie HTML in diesem Feld verwenden. Denn wenn ein HTML-Tag nicht abgeschlossen wird, kann die ganze Darstellung chaotisch sein.	100
specialPreparation	string	Dieser Text wird Ihre ###PRODUCT_SPECIAL_PREP### Markierungen ersetzen.	
createUsers	Boolean	Wenn Sie das und PIDuserFolder setzen, dann wird für jeden Kunden, der nicht eingeloggt ist, und dessen Email-Adresse noch nicht als Benutzer gespeichert wurde, als neuer Frontend Benutzer erzeugt. Eine Email mit dem Benutzernamen (seine E-mail Adresse) und dem Paßwort wird ihm zugesendet. (siehe ###EMAIL_NEWUSER_TEMPLATE###)	0
outputFolder	string	Verzeichnis, wo die Rechnung und der Lieferschein angelegt werden.	fileadmin/data
memberOfGroup	int+	Die ID der Frontend Benutzergruppe, der die neu erzeugten Frontend Benutzer angehören.	1
discountGroupName	string	Name der Frontend Gruppe für einen Discount Preis	
AGBtarget	small	Das Target-Attribut im Link einer PIDagb	_blank
AGBattachment	resource	Wenn diese Datei existiert, wird sie an die Email der Bestellbestätigung an den Kunden als Anhang mitgesendet. Der Shop Administrator wird keinen AGB als Anhang per Email erhalten, aber sehr wohl eine CSV Datei mit den Bestelldaten, wenn Sie das konfiguriert haben.	Fileadmin/agb.pdf
GiftAttachment	resource	Wenn diese Datei existiert, wird sie an die Email zum Versenden eines Geschenkgutscheins angehängt.	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
generateCSV	boolean	Erzeugt aus jeder Bestellung eine CSV-Datei. Diese Datei wird auf dem Server abgespeichert und per Email an den Shop Administrator versendet (mit einer Bestellbestätigung).	0
CSVdestination	../csvorders/	Das Verzeichnis auf dem Server, wohin die CSV Dateien abgespeichert werden. Dieses Verzeichnis sollte nicht über Http zugänglich sein, weil sonst ein jeder die Daten Ihrer Kunden und der von ihnen gemachten Bestellungen einsehen könnte. Also stellen Sie dieses Verzeichnis nicht unter den Pfad von fileadmin, oder geben Sie ihm wenigstens einen schwer zu erratenden Namen.	../csvorders/
CSVfields	string	Angabe, welche Produktdaten als CSV gespeichert werden sollen. Standardmäßig werden alle Felder bis auf Bemerkung verwendet.	(Long, long list)
CSVinOneLine	boolean	Wenn Sie das auf 1 setzen, werden zusätzliche Informationen (Versand, Bezahlung, Adreßdaten, Mitteilung zur Lieferung) an den ersten bestellten Artikel angehängt. Das vereinfacht den Import dieser Dateien. Wenn Sie das auf 0 gestellt lassen, wird diese Datei in Programmen wie Excel am besten angezeigt.	0
alwaysInStock	boolean	Wenn gesetzt, wird die Anzahl der Produkte auf Lager nach einem Kauf nicht reduziert.	1
showNotinStock	boolean	Wenn gesetzt, dann werden die Produkte auch dann angezeigt, wenn sie nicht auf Lager sind.	
notInStockMessage	string	Siehe oben. Das wird angezeigt, wenn keine Produkte auf Lager sind und wenn showNotinStock gesetzt ist.	Not in stock
warningInStockLimit	int+	Anzahl von Produkten auf Lager, ab welcher eine Warnemail verschickt wird, sobald sie erreicht wird.	
inStockPieces	string	Das ist die Einheit für Produkte auf Lager inStock.	pieces
newItemDays	int+	In LISTNEWITEMS werden die in den letzten n Tagen neu hinzugekommenen Artikel angezeigt.	7
bulkilyWarning	string	Text für ###BULKILY_WARNING### für sperrige Produkte	
bulkilyAddition	int+	Betrag um den die Versandkosten für ein sperriges Produkt erhöht werden.	
bulkilyFeeTax	int+	Steuerbetrag in Prozent für das Versenden eines Sperrguts	

javaScript	*array of integers*	verschiedene JavaScripts, die für ###JAVASCRIPT_10### Markers eingefügt werden
		
		Beispiel:
		`javaScript {`
		`  10.value = function addValues (a, b) { return a+b; }`
		`  20.value = function multiplyValues (a, b) { return a*b; }`
		`}`	

payment / shipping	(siehe unten)	Konfiguration der Bezahlungsarten und Liefermethoden. Hier werden die Textausgaben, die Kosten und zusätzliche Berechnungen und Einstellungen für Bezahlsysteme eingestellt. Siehe Beschreibung weiter unten!	

[tsref:(script).class.tx_ttproducts_pi1.php]

CSS Konfiguration

Hier können die CSS id Namen gesetzt werden. Sie müssen eine CSS Datei im Template Setup angegeben haben, die diese ids auch verwendet.

Sie müssen auch den Namen der Tabelle im Setup mit angeben.

```
plugin.tt_products.CSS.tt_products.row.even = 35
```

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
row		even: Cascading Stylesheets (CSS) gerade Zeilen in der Produktliste. uneven: ungerade	
list		default: CSS für Standardeinträge in der Listenansicht current: CSS für die ausgewählten Einträge in der Listenansicht.	
menu	string	CSS für das Menü	

Konfiguration der Artikel, Produkte, Kategorien, Seiten und Bilder

Der vor/zweitletzte Wert enthält den Codenamen der Ansicht. Wenn hier 'ALL' steht, dann gilt es für alle Ansichten.

Ansichten, die dem Code-Feld entsprechen:

SINGLE, LIST, BASKET

zusätzlich mögliche Werte:

EMAIL

Beispiel:

```
plugin.tt_products.conf.tt_products_articles.LIST.generatePath.base = fileadmin/images
plugin.tt_products.conf.tt_products.LIST.orderBy = sorting
```

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung:

generatePath	string	Pfad zu den Verzeichnissen der Bilder für generateImage
		
		Paare von Feldnamen und die Anzahl der ersten Buchstaben, die zur Bildung des Namens der Bilddatei verwendet werden.
		type ... tablefields (Tabellenfelder)
		field.*fieldname* ... Name des Tabellenfeldes
		
		Beispiel:
		`ALL.generatePath {`
		`  type = tablefields`
		`  base = fileadmin/images`
		`  field.itemnumber = 2`
		`}`
			fileadmin/img

generateImage	*array of fields*	Paare von Feldnamen und die Anzahl der ersten Buchstaben zur Bestimmung des Namens der Bilddatei.
		
		type ... tablefields (Tabellenfelder), foreigntable (andere Tabelle, falls benötigt)
		field.*fieldname* ... Name des Feldes der Tabelle
		table ... andere Tabelle und deren Konfiguration um das Bild zu ermitteln
		uid_local ... verwende dieses lokale Feld der Tabelle
		uid_foreign ... verwende dieses Feld der externen Tabelle um eine Übereinstimmung zu finden
		parent ... verwende auch Feldnamen der Vater-Tabelle
		parent.depth ... Anzahl der Vaterebenen
		
		Beispiel:
		`ALL.generateImage {`
		`  type = tablefields`
		`  field.itemnumber = 6`
		`}`
		
		`ALL.generateImage {`
		`  type = foreigntable`
		`  table = tt_products_articles`
		`  uid_local = uid`
		`  uid_foreign = pid`
		`  field.itemnumber = 6`
		`}`	

imageMarker		Beschreibt woraus sich der Marker für das Bild zusammensetzt. Als Beispiel wird das Bild der Datei 30_P1_001.jpg verwendet. Der zweite Teil P1 und der dritte Teil 001 bilden den Marker. Als Ergebnis erhält man den Marker ###CATEGORY_IMAGE_P1_001###. Der erste Teil '30' muß in diesem Beispiel pro Datensatz verschieden sein.
		
		Beispiel:
		`ALL.imageMarker {`
		`  type = imagename`
		`  parts = 2,3`
		`}`	

orderBy	string	Liste der Felder, nach denen die Artikel sortiert werden.	sorting

fetchImage	string	Die Bilder einer anderen Tabelle werden statt den Bildern dieser Tabelle verwendet.
		
		Example:
		`plugin.tt_products.conf.tt_products.ALL.fetchImage {`
		`  type = foreigntable`
		`  table = tt_content`
		`}`	

Eigenschaft:	Datentyp	Beschreibung:	Voreinstellung:

language		Der Name der Sprachdatei mit Übersetzungen aus der Default Sprache in eine andere Sprache.
		type:
		❿ csv ... Die Werte werden mit ';' und Zeilenumbrüchen getrennt
		❿ noTranslation ... die Tabelle der alternativen Sprache soll nicht verwendet werden
		❿ field ... Die Übersetzung befindet sich in Feldern
		
		file: Pfad und Name der Datei
		
		Beispiel:
		`[globalVar = GP:L = 1]`
		`language {`
		`type = csv`
		`file = fileadmin/data/EnglishCategories.csv`
		`}`
		`[GLOBAL]`
		
		Beispiel:
		`language {`
		`type = field`
		`field.title = subtitle`
		`}`	

image	IMAGE cObject	Das Image Objekt kann über TypoScript in andere kopiert werden und somit für mehrere Code Felder bzw. Ansichtsarten verwendet werden. **Beispiel:** `###PRODUCT_IMAGE1:M###` `plugin.tt_products.conf.tt_products.ALL.image.m {` `wrap = | ` `file.maxW = 320` `file.maxH = 280` `}`	

Formular Einstellungen

Nur für Setup. Es gibt mehrere Formulare, die konfiguriert werden können. Geben Sie den Code nach dem 'form' an.

Example:
`plugin.tt_products.form.SEARCH.name = ShopSearchForm`

Property:	Data type:	Description:	Default:
name		Name des Formulars. Falls leer, wird der gesamte Subpart (Unterbereich) nicht gezeichnet.	*abhängig vom Codefeld*

Konfiguration für Bezahlung und Versand

Die Bezahlung und der Versand sind einander von der Konfiguration her sehr ähnlich. Daher haben sie im wesentlichen dieselbe Liste an Eigenschaften mit Anmerkungen, wenn eine Eigenschaft nur bei einem zutrifft. Der Konfiguration der Bezahlung und des Versands bietet die Möglichkeit, verschiedene Versandarten und Bezahlmethoden für den Benutzer anzugeben. Außerdem können Sie festlegen, ob es über Auswahlknöpfe oder eine Auswahlbox wählbar sein soll.

Die Nummer der ausgewählten Bezahl- und Versandmethode wird am Ende der Template Datei in der Dankeschön-Seite wieder ausgewertet, damit nur der entsprechende Text dort aufscheint. Sie können mit diesen Einstellungen auch die Ausführung von besonderen PHP Skripten veranlassen. Das ist alles möglich.

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
radio	boolean	Wenn gesetzt, werden Auswahlknöpfe angezeigt, ansonsten eine Auswahlbox.	0
template	string	**(Nur Layout Auswahlknöpfe)** Wenn .radio gesetzt ist, dann wird dieser Text als 'template' für die Auswahlknöpfe verwendet. **Standard ist (in einer Zeile):** `<nobr>###IMAGE### <input type="radio" name="recs [tt_products]['.$key.']" onClick="submit()" value="###VALUE###"###CHECKED###> ###TITLE###</nobr> `	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
wrap	string	**(Nur Auswahlbox Layout)** Wenn .radio 0 ist, dann wird dieser Text um die <option> Tags in einem <select>-Tag ausgegeben. **Standard ist (in einer Zeile):** `'<select name="recs[tt_products][' .$key. ']" onChange="submit()">|</select>'`	

TAXpercentage	double	Steuersatz.
		Double Wert (!) (das bedeutet, daß der Punkt '.' für die Dezimaltrennung verwendet werden muß)
		Das ersetzt die frühere Eigenschaft priceNoTax. Dieser Steuersatz kann sich von dem globalen mit dem gleichen Namen unterscheiden.
		
		Beispiel:
		`# Dänischer Steuersatz beträgt 25%:`
		`payment.TAXpercentage = 25.00`
			global übernommen

TAXincluded	boolean	Setzen Sie das, wenn die Steuer in den Preisen für Bezahlung/Versand miteingerechnet ist.	global übernommen

Feld von Ganzzahlen		Konfiguration, siehe unten
		
		Beispiels:
		`TAXpercentage = 12`
		`10.title = Credit card`
		`10.image.file = media/logos/dankort.gif`
		`10.price =`
		`10.percentOfGoodstotal = 0`
		`10.calculationScript =`
		`media/scripts/products_comp_calcScript.inc`
		
		`30.title = By mail`
		`30.image.file = media/logos/postdanmark.gif`
		`30.price = 40`	

Konfiguration der Einträge zur Bezahlung / Versand			
title	string	Name des Eintrags, z.B. “Master Card” oder “Postversand” Der Name wird mehrmals hintereinander aufgelistet, wenn Marker ###STATIC_COUNTRIES_CN_ISO_3### vorkommen und where.static_countries gesetzt worden ist.	
image	IMAGE cObject	Logo Bild für den Eintrag	

price	double oder *Feld von Ganzzahlen siehe unten für weiter Parameter*	Preis eines Eintrags, inkl. oder exkl. Steuer je nach TAXincluded.
		Verwenden Sie Ganzzahlen, um die minimale Anzahl an Artikeln festzulegen, für die der Preis gültig ist. 6 Artikel und mehr würden in diesem Beispiel 5.8 Euro kosten.
		Mit dem Typ weight können Sie auch die Versandkosten nach Gewicht der Produkte berechnen.
		
		Beispiel:
		`30.price.type = count`
		`30.price.1 = 4`
		`30.price.6 = 5.8`	

~~priceTax~~	double	wird eingestellt (1st July 2008) Preis eines Artikels inkl. Steuer	verwenden Sie price, TAXpercentage und TAXincluded
~~priceNoTax~~	double	wird eingestellt (1st July 2008) Preis eines Artikels ohne Steuer Achtung: hier müssen Sie den Steuersatz selber berechnen!	verwenden Sie price, TAXpercentage und TAXincluded
replaceTAXpercentage	double	siehe TAXpercentage Wenn gesetzt, wird die allgemeine TAXpercentage damit überschrieben.	
priceFactWeight	double	**nur Versand:** Der Preis wird aus dem Gewicht aller Produkte berechnet. Das wird zum Gesamtpreis dazugezählt. Das Gewicht wird mit diesem Faktor multipliziert, um den Versandpreis zu erhöhen.	
getDiscountPrice	boolean	Wenn gesetzt, wird der Discount Preis auch für normale Kunden gewährt, ohne daß sie zu einer Frontend Benutzergruppe in einer Discount Gruppe gehören müssen.	
percentOfGoodstotal	double	Preis eines Artikels, berechnet über einen Prozentsatz des Gesamtbetrags vor der Bezahlung/Versand	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
percentOfTotalShipping	double	**nur Versand:** Wenn gesetzt, werden die Kosten für das Bezahlsystem als Prozente vom gesamten Produktpreis inklusive der Versandkosten und der Steuer errechnet.	
calculationScript	resouce	PHP Skript, das inkludiert wird. Es soll die internen Arrays zur Berechnung heranziehen. Dieses Skript kann dazu verwendet werden, um eine besondere Gebühr für Bezahlung/Versand zu berechnen. Für eine Beispielsanwendung siehe pi/products_comp_calcScript.inc, das zeigt wie der Endbetrag um 5.75% seines eigenen Wertes erhöht wird, um die Gebühren der internationalen Kreditkarten Organisation abzudecken. Das Berechnungsskript erhält die TypoScript Eigenschaften im $conf Array.	
handleScript	resource	PHP Skript, das nach dem Abschluß einer Bestellung aufgerufen wird. Dieses Skript muß sich selber um die Bildschirmausgabe und um den Abschluß der Bestellung kümmern. Siehe pi/payment_DIBS.php als Beispiel. Die Eigenschaften des Handle Skripts werden an die Funktion als $conf Array übergeben. Der Inhalt der Variable $content muß den HTML-Inhalt als Ergebnis zurückliefern.	
handleLib	string	**nur Bezahlung:** Name der TYPO3 Library für das Bezahlungssystem. Im Moment kann hier nur 'paymentlib' gesetzt werden, um die Payment Library Extension von Rober Lemke zu verwenden. **Beispiel:** `30.handleLib = paymentlib`	
handleURL	string	Wenn gesetzt, wird dieses handleURL anstelle der Dankeschön-Seite (über PIDthanks) ausgegeben, damit z.B. ein handleScript die Information bzgl. Bezahlsystem usw. Verarbeiten kann.	
handleTarget	string	Alternatives Target der HTML-Form.	

excludePayment	*list of integers*	**nur Versand:**
		Dies ist eine Liste an Schlüsseln der Bezahlmethoden (ihre Nummern), die für eine bestimmte Versandart nicht angezeigt werden. Wenn zum Beispiel Leute Waren im Geschäft kaufen, dann möchten Sie nicht, daß diese Geld überweisen oder Online bezahlen, sondern daß sie die Waren sofort bar bezahlen. Daher können Sie diese Bezahlmethoden ausschließen.
		
		Beispiel:
		`...`
		`  40.title = Abholen der Waren im Geschäft`
		`  40.excludePayment = 10,40`
		`}`	

replacePayment	*list of integers*	**nur Versand:**
		Diese Einstellungen der Bezahlung werden überschrieben, wenn diese Bezahlmethode ausgewählt worden ist.
		
		Example:
		`...`
		`  40.title = China`
		`  40.replacePayment.10.title = Bezahlung nach China`
		`  40.replacePayment.10.price = 100`
		`}`	

show	boolean	Wenn das gesetzt ist, wird der Eintrag in der Liste angezeigt.	1
showLimit	double	Wenn gesetzt, dann wird der Eintrag nur dan angezeigt, wenn zumindest diese Anzahl an Produkten im Warenkorb liegt. 0 ... zeige diesen Eintrag immer an	0
type	string	**nur Bezahlung:** fe_users ... die Bezahlart kann über die fe_users Tabelle konfiguriert werden	
visibleForGroupID	int+	**nur Bezahlung:** Diese Bezahlmethode ist nur verfügbar, wenn ein Benutzer eingeloggt ist und wenn er ein Mitglied dieser Benutzergruppe ist.	
addRequiredInfoFields	string	**nur Bezahlung:** Zusätzlich benötigte Felder auf der INFO Seite, wenn diese Bezahlmethode ausgewählt worden ist. Brauchbar für Bezahlungen mit der Kreditkarte.	
Configuration price Parameters for payment / shipping			
type	string	Bedeutung der Nummer: count ... die Anzahl der Produkte weight ... das berechnete Gewicht in Gramm price ... der Gesamtpreis der Produkte	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

WherePIDMinPrice	*list of integers*	**nur Versand:**
		Setzen eines Minimalpreises für den Versand, wenn es einen Artikel im Warenkorb gibt, der aus dem Sysfolder mit der angegebenen PID stammt. Wobei 155 die PID ist und 7.5 ist der Minimalpreis, der als Versandkosten berechnet wird, wenn zumindest ein Artikel im Warenkorb mit der PID liegt.
		
		Beispiel:
		`plugin.tt_products.shipping {`
		`10.title = Paket`
		`10.price.type = weight`
		`10.price.WherePIDMinPrice.155 = 7.5`
		`10.price.1 = 1.5`
		`10.price.500 = 2.5`
		`10.price.1000 =3.5`
		`}`	

where.static_countries	string	**nur Versand:**
		Zum Setzen einer SQL WHERE Bedingung für die mit der Select-Box ausgewählten Länder. Diese stammen aus der static_countries Tabelle der static_info_tables Extension.
		
		Beispiel:
		`plugin.tt_products.shipping {`
		`10.title = Paket Deutschland`
		`10.where.static_countries = cn_short_local == 'Deutschland'`
		`10.price = 5.9`
		`20.title = Paket EU`
		`20.where.static_countries = cn_eu_member = 1 AND cn_uno_member = 1 AND cn_short_local != 'Deutschland'`
		`20.price = 8.9`
		`30.title = außerhalb EU`
		`30.where.static_countries = cn_short_local == 'Deutschland'`
		`30.price = 15`
		`}`	

~~productsNoTax~~	~~int+~~	~~**nur Versand:**~~ ~~Wenn gesetzt wird für kein Produkt eine MWSt verrechnet.~~	

noCostsAmount	double	Wenn der Gesamtpreis für ein Produkt diese Summe erreicht hat, dann werden keine Kosten berechnet.
		
		Beispiel:
		`plugin.tt_products.shipping {`
		`10.title = Paket`
		`10.price.type = weight`
		`10.price.noCostsAmount = 200`
		`10.price.1 = 1.5`
		`}`	

Pricecalc, discountprice und creditpoints Konfiguration

Die Preisberechnung (pricecalc) gibt Ihnen die Möglichkeit, Preissummen über die Produkte über eine Berechnungstabelle zu ermitteln. Der Discount Preis (discountprice) wird für Kunden verwendet, die Frontend Benutzer sind und zur Gruppe mit dem Namen gehören, der über discountGroupName gesetzt worden ist..

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
prod	*Zweischneidige Liste von Ganzzahlen*	Die linke Seite der Integer kennzeichnet Zeilen, die zusammengehören. Die Bedeutung der rechten Integer ergibt sich aus den Einstellungen jeder Zeile.	0

		Pricecalc:
		Spezielle Preise für die Produkte. Wo 1 Produkt 4.99 kostet, dort werden 2 Produkte 8.99 kosten. Mit discountprice wird der Preis für ein Produkt eingegeben, mit pricecalc ist es der Preis für alle Produkte zusammen, für die eines davon 4.99 als Preis im Sysfolder eines x-beliebigen Produktes eingetragen ist. Der discountprice überschreibt die Ergebnisse von pricecalc, weil hier immer das niedrigere Ergebnis als Endpreis herauskommen muß. Wenn das Preisfeld price2 verwendet wird, dann überschreibt dieses alle Preisberechnungen.
		
		Beispiel:
		`pricecalc {`
		`  10.type = count`
		`  10.field = price`
		`  10.where =`
		`  10.prod.1 = 4.99`
		`  10.prod.2 = 8.99`
		`  10.prod.5 = 19.99`
		`  20.type = count`
		`  20.field = price`
		`  20.where =`
		`  20.prod.1 = 6.99`
		`  20.prod.2 = 13.98`
		`  20.prod.5 = 29.99`
		`}`	

		Discountprice:
		Hier werden die Preise der Produkte entsprechend der Typ-Einstellung berechnet. Bei 'count' werden die Preise der Artikel zusammengezählt. Die 'additive'-Einstellung bewirkt, daß die Artikel über alle Konfigurations-Zeilen hinweg zusammengezählt werden und nicht wie sonst, nur zeilenweise.
		
		Beispiel:
		`discountprice {`
		`  10.type = count`
		`  10.field = price`
		`  10.additive = 1`
		`  10.where =`
		`  10.prod.1 = 4.99`
		`  10.prod.100 = 2.49`
		`  10.prod.1050 = 2.39`
		`  20.type = count`
		`  20.field = price`
		`  20.where =`
		`  20.prod.1 = 6.99`
		`  20.prod.100 = 2.59`
		`  20.prod.1050 = 2.49`
		`}`	

		Creditpoins:
		Damit wird eingestellt, wieviele Gutscheinpunkte jemand erhalten wird, wenn er Artikel im Shop kauft. Die Werte auf der rechten Seite sind die Prozente der bestellten Artikel, wenn 'type=price' ist.
		
		Beispiel:
		`creditpoints {`
		`  pricefactor = 0.5`
		`  10.type = price`
		`  10.prod.1   = 0.02`
		`  10.prod.101 = 0.04`
		`  10.prod.501 = 0.06`
		`}`	

additive	double	Nur für discountprice gültig. Wenn gesetzt, werden alle Produkte von allen Discount-Preisen zusammengezählt, um den effektiven Disount-Preis zu berechnen. Wenn nicht gesezt, dann werden nur die Produkte mit demselben Preis zusammengezählt.	
type	string	Die Bedeutung der Ganzzahl auf der rechten Seite, die normalerweise zusammengezählt wird: count ... die Anzahl der Produkte (nur pricecalc und discountprice) price ... das verwendete Preisfeld (nur creditpoints)	

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:

pricefactor	double	Damit wird berechnet, wieviel Geld jemand für seine Gutscheinpunkte erhalten wird. 2 Gutscheinpunkte werden 1 Euro an Gutschrift in der eingestellten Währung ergeben.
		
		Beispiel:
		`creditpoints {`
		`  pricefactor = 0.5`
		`}`	

[tsref:(script).productsLib.payment/(script).productsLib.shipping]

payment_DIBS.php

Eigenschaften des handleScript für eine Schnittstelle zu DIBS (http://www.architrade.com/uk/):

Eigenschaft:	Datentyp:	Beschreibung:	Voreinstellung:
templateFile	resource	Template Datei für die Verwendung mit DIBS Sie müssen folgende Zeile in die FORM des tt_products Templates eintragen, bevor das DIBS Skript aufgerufen wird: `<input type="hidden" name="products_cmd" value="cardno">`	
soloe	boolean	Wenn gesetzt, verwendet dieses Skript Unter-Templates mit der Markierung ###DIBS_SOLOE_TEMPLATE### anstelle der standardmäßigen Markierung ###DIBS_CARDNO_TEMPLATE###	
direct	boolean	Wenn gesetzt, verwendet dieses Skript Unter-Templates mit der Markierung ###DIBS_DIRECT_TEMPLATE### anstelle der standardmäßigen Markierung ###DIBS_CARDNO_TEMPLATE###	
merchant	boolean	Händler id	
currency	int+	Währungsnummer, ISO4217 Format	
relayURL	string	Die Url des Shop dessen gesicherter Server den Warenkorb übernimmt.	
test	boolean	Wenn gesetzt, wird das Test-Feld in der FORM gesetzt.	

cardType	string	Kartentyp,
		
		Beispielwerte:
		`DK = Dankort`
		`V-DK = Visa-Dankort`
		`MC(DK) = Mastercard/Eurocard issued in Danmark`
		`VISA = Visakort issued abroad`
		`MC = Mastercard/Eurocard issued abroad`
		`DIN(DK) = Diners Club, Denmark`
		`DIN = Diners Club, international`	

account	string	DIBS Kontoeigenschaft	
addOrderInfo	boolean	Wenn gesetzt, werden die Bestelldaten an die FORM angehängt. DIBS kann diese Information auswerten und zusammen mit den Bezahlinformationen anzeigen.	
k1 k2	string	DIBS Schlüsselwerte	

[tsref:(script).productsLib.paymentDIBS]

Tutorial

Wenn Sie noch ein Anfänger mit dem Shop System sind, dann sollten Sie mit einer Schritt-für-Schritt Anleitung beginnen, die Sie unter dem Extension Key *tut_ttproducts_de* aus dem TER herunterladen können.

Beispielkonfiguration von Inter-Photo A/S (www.inter-photo.dk):

```
xxxxx.xxxx {
  10.title = Dankort, VISA-Dankort
  10.image.file = media/logos/dankort.gif
  10.image.params = align=absmiddle vspace=2
  10.handleScript = media/scripts/payment_DIBS.php
  10.handleScript {
    merchant = xxxxx
    test = 0
    k1 = xxxxx
    k2 = xxxxx
    currency = 208
    addOrderInfo = 1
    account =
    cardType = DK,V-DK
    relayURL = http://www.inter-photo.dk/index.php?id=204
  }
  10.handleURL = index.php?id=204
  10.handleTarget = _top

  20 < .10
  20.title = Unibank e-betaling
  20.image.file = media/logos/soloe.gif
  20.handleScript.soloe=1
}
```

Beispiel der Produktansicht, damit der Titel eines Produkts zum Seitentitel wird

Wenn Sie den Titel eines Produkts in eine Markierung einlesen wollen, z.B. für die Verwendung mit automake_template oder Ähnlichem, dann probieren Sie diesen Code aus:

Auszug aus dem Code:

```
# setzen Sie zuerst den Seitentitel auf den Titel der aktuellen Seite
temp.pagetitle = TEXT
temp.pagetitle.field = title

# jetzt überschreiben Sie den mit dem Titel des Produkts, wenn das Feld nicht leer ist
[globalVar = GP:tt_products > 0]
temp.pagetitle = COA
temp.pagetitle {
    10 = RECORDS
    10 {
        source.data = GPvar:tt_products
        tables = tt_products
        conf.tt_products = TEXT
        conf.tt_products {
           field = title
        }
    }
}
// Für die Einzelansicht eines Datensatzes muß der Cache ausgeschaltet werden.
config.no_cache = 1
[global]

temp.mainTemplate = TEMPLATE
temp.mainTemplate {
    # Einspeisen des Inhalts aus dem Auto-Parser in das TEMPLATE cObject:
    template =< plugin.tx_automaketemplate_pi1
    # Wähle nur den Inhalt zwischen den <body>-Tags
    workOnSubpart = DOCUMENT_BODY
    subparts.title < temp.pagetitle
```

....

Beispiel Templates

Hier kommt der Teil eines Templates, wenn der Benutzer automatisch als Frontend Benutzer registriert worden ist.

```
<BR><BR><BR><BR><HR><BR><BR><BR> <h3>EMAIL_NEWUSER_TEMPLATE</h3> <em> Teil, der als Template für die
Emails zur Verständigung über das Anlegen eines Benutzers verschickt wird. Die erste Zeile wird in die
Betreff-Zeile der Email eingetragen.<br> </em><br> <pre> <!-- ###EMAIL_NEWUSER_TEMPLATE### begin --> Es
wurde ein neuer Benutzer angelegt. Sehr geehrter Herr ###PERSON_NAME###, Sie haben zum ersten Mal eine
```

```
Bestellung auf http://...../ getätigt. Um Ihnen in Zukunft das Bestellen zu erleichtern,
haben wir ein Benutzerkonto für Sie eingerichtet. Ihre Benutzerkonto Daten: Benutzername: ###USERNAME###
Paßwort: ###PASSWORD###
 Mit freundlichen Grüßen,
 Ihr Shopmaster <!-- ###EMAIL_NEWUSER_TEMPLATE### end --> </pre>
```

Beispiel bananaGuard

(siehe das deutsche example_template_bill_de.tmpl von http://bananaguard.de)

Dieses Beispiel beinhaltet spezielle Preisberechnungen und die automatische Erzeugung einer Rechnung und eines Lieferscheins. Sie müssen die TYPO3-Erweiterungen feuser_admin und conf_userts installieren, wenn Sie ein ähnliches Shop aufbauen möchten. Fertigen Sie eine Kopie des example_template_bill_de.tmpl in fileadmin/tmpl_files/products_eur_.tmp an. Damit Sie dieses Template verwenden können, müssen Sie nach '?id=' im Text suchen und dahinter die PIDs des Warenkorbs mit jener auf Ihrem System ersetzen.

Die Seiten und Plugins auf dem Seitenbaum sind:

- 'BananaGuard' mit dem Haupttemplate Ihrer Wahl und

Constants:

```
plugin.tx_srfeuserregister_pi1.email = info@shopms.de
plugin.tx_srfeuserregister_pi1.confirmPID = 83 ... Sie müssen Ihre confirm PID hier eintragen.
styles.content.loginform.pid = 108 ... Die PID des Benutzer sysfolder.
```

-- 'Home', das eine Verknüpfung zu BananaGuard ist

-- 'Preise und Versand' mit speziellen Informationen über Preise und Versand

-- 'BananaShop' mit einem eigenen Template für das Shop

Constants:

```
plugin.tt_products.file.templateFile = typo3/ext/tt_products/pi/products_eur_.tmpl
plugin.tt_products.TAXpercentage = 16
plugin.tt_products.priceDecPoint = .
plugin.tt_products.maxW_list = 80
plugin.tt_products.outputfolder = fileadmin/data

plugin.tt_products.color2 = #003399
plugin.tt_products.color1 = #FFFFFF
plugin.tt_products.wrap2 = <font face="verdana" size=2 color=white> | </font>
content.tableCellColor = #003399
```

Setup:

```
plugin.tt_products.code.field = select_key
plugin.tt_products.alwaysInStock = 1

plugin.tt_products.statusCodes.1 = Bestellungseingang
plugin.tt_products.statusCodes.11 = Der bananaSHOP wartet auf Ihren Zahlungseingang
plugin.tt_products.statusCodes.20 = Ihre Ware wird versendet
plugin.tt_products.statusCodes.101 = Bestellung abgeschlossen
plugin.tt_products.statusCodes.200 = Bestellung storniert

plugin.tt_products.orderEmail_from = info@bananaguard.de
plugin.tt_products.orderEmail_fromName = bananaGUARD.de
plugin.tt_products.orderEmail_to = info@bananaguard.de
orderNumberPrefix = order2005_

plugin.tt_products.discountGroupName = Team
plugin.tt_products.lockLoginUserInfo=true

plugin.tt_products.displayBasketColumns = 3
plugin.tt_products.outputFolder = {$plugin.tt_products.outputfolder}

plugin.tt_products {
  payment >

payment {
  radio = 1
  TAXpercentage = 16
  10.title = Vorkasse
  20.title = PayPal
  20.percentOfTotalShipping = 0.04
  30.title = Nachnahme
  30.discountDeactive = 1
  30.price.1 = 4
  30.price.6 = 5.8
  30.showLimit = 99
}

shipping {
  radio = 1
  TAXpercentage = 16
  10.title = Deutschland
  10.image.file =
  10.price.type = count
```

```
  10.price.1 = 2.5
  10.price.6 = 4.8
  10.price.50 = 10
  10.price.100 = 25
  10.price.120 = 30
  10.price.300 = 150
  10.percentOfGoodstotal = 0

  20.title = Europa (nicht Deutschland)
  20.image.file =
  20.price.type = count
  20.price.1 = 8.8
  20.price.6 = 11.8
  20.price.50 = 11.8
  20.price.100 = 25
  20.price.120 = 30
  20.price.300 = 150
  20.percentOfGoodstotal = 0

  30.title = Selbstabholung
  30.image.file =
  30.price.1 = 0
  30.percentOfGoodstotal = 0
}

pricecalc {
  type = count
  field = price
  10.prod.1 = 4.99
  10.prod.2 = 8.99
  10.prod.5 = 19.99
  20.prod.1 = 6.99
  20.prod.2 = 13.98
  20.prod.5 = 29.99
}

discountprice {
10.type = count
10.field = price 10.additive = 1
10.where =
10.prod.1 = 4.99
10.prod.100 = 2.89
10.prod.1050 = 2.77
20.type = count
20.field = price 20.where =
20.prod.1 = 6.99
20.prod.100 = 3.00
20.prod.1050 = 2.89
}
}

plugin.tt_products.basketImage.imageLinkWrap.height = 800
plugin.tt_products.listImage.file.maxW = 150
plugin.tt_products.shipping.40 >
plugin.tt_products.PIDagb =
plugin.tt_products.createUsers = 1
plugin.tt_products.orderEmail_htmlmail = 1
```

Plugin:

Produkte , CODE: LIST

Die nächsten Seiten sind:

--- 'Warenkorb'

Plugin:

Produkte, CODE: BASKET, Auf dem Rand ein Inhaltselement 'Login' das auf die Seite 'Warenkorb' verzweigt

--- 'Kasse'

Plugin:

Produkte , CODE: PAYMENT, FINALIZE, INFO

--- 'Bestellungsstatus'

Plugin:

Produkte , CODE: TRACKING

--- 'Rechnung'

Plugin:

Produkte , CODE: BILL

--- 'Lieferung'

Plugin:

Produkte, CODE: DELIVERY

--- 'Articles' Sysfolder

Plugin:

Produkte, CODE: LIST

-- 'Kontakt/Bestellung' beinhaltet eine Form, Text und ein Login

-- 'BananaINSIDER' nur nach dem Frontend Login sichtbar, beinhaltet eine Login Plugin auf dem rechten Rand

Die Gruppe 'Team' hat Zugriff auf diese Seite.

--- 'My profile'

Plugin:

Front End Benutzer Registrierung

--- 'INSIDER Infos'

--- 'bananaBOARD'

Plugin:

Board, Tree, CODE: FORUM, POSTFORM

-- 'Users' Sysfolder der Gruppe 'Team'

Bekannte Probleme

Prüfliste:

- füge unter 'Template' – Static (aus Extensions) eine statische Template Datei 'Shop System' ein.
- **niemals** das 'Shop System Test' oder das 'plugin.tt_producs [DEPRECATED]' verwenden
- setze 'auf Lager' für alle Produkte und Artikel auf einen Wert größer 0
- verwende keine TYPO3-Einstellungen zur Mehrsprachigkeit oder lege Einträge in den Tabellen 'Produkte – alternative Sprache' an.
- die Shop Template Datei muß gefunden werden
- ein Fehler im Shop erzeugt einen Eintrag in der PHP error_log Datei (im Install Tool und php.ini aktivierbar)
- der Cache ist vorher gelöscht worden

Allgemein:

- Wenn Sie über den Extension Manager ein Update von einer vorher installierten älteren Version von tt_products machen, und wenn Sie vorher nicht die Table Library 'table' und FEBE Library 'fh_library' installiert haben, dann wird das ganze TYPO3 CMS unbenutzbar und Sie erhalten ide folgende Fehlermeldung

  TYPO3 Fatal Error: Extension key "table" was NOT loaded! (t3lib_extMgm::extPath)

  Schritte zur Reparatur des TYPO3 CMS:

  ```
  1. Editieren Sie die Datei typo3conf/localconf.php
  2. Löschen Sie den Eintrag ',tt_products'
  3. Löschen Sie die temp_CACHED_ Dateien aus typo3conf
  4. Klicken Sie auf Neuladen im Internet Browser
  ```

- Bitte sehen Sie auf die Webseite http://wiki.typo3.org/index.php/Ext_tt_products . - Laden Sie die letzte Entwicklungsversion von http://fholzinger.com/development/tt_products-2.5.2.<subnr>.tar.gz . <subnr> ... Hier müssen Sie selber eine Unternummer zur aktuellen Version im TER ausprobieren.
  Oder schauen Sie auf http://fholzinger.com nach.

- .htaccess muß richtig konfiguriert werden, sonst läuft tt_products nicht wie erwartet, weil das Plugin sonst nicht alle benötigten Skripte finden kann.

- **Falscher Parameter, GET/POST var 'tx_ttproducts_pi1[product]' wurde nicht angegeben oder kein Produkt mit uid = 0 gefunden.**

  Sie müssen die Einstellungen für PIDitemDisplay vornehmen, damit der Shop weiß, wie der Linkparameter 'tx_ttproducts_pi1[product]' auf die Einzelansicht erzeugt werden muß.

- **Rechtliches**: http://www.wettbewerbszentrale.de/media/getlivedoc.aspx?id=567

Fehler im Online-Handel: http://www.wettbewerbszentrale.de/media/getlivedoc.aspx?id=568
Beim Versand ins Ausland müssen Sie auch die Gesetze aller Länder genauestens beachten, in die Sie Waren versenden. Damit ersparen Sie sich sehr hohe Kosten durch Abmahnungen von Anwälten eines Konkurrenzunternehmers. Im Internet kann der Abmahner den Gerichtsort bestimmen.

- Das Plugin zeigt eine leere Seite
  Siehe die Prüfliste oben.

To-Do Liste

- Wunschliste: http://wiki.typo3.org/index.php/Ext_tt_products#Wishlist
- Umschreiben des Codes für PHP5.

Changelog

siehe auch Datei tt_products/ChangeLog

- 26.03.2005 Einbringen des Codes von Klaus Zierer (zk_products), mehrspaltige Listenansicht, Rechnung, Lieferschein und spezielle Preisberechnungen von Franz Holzinger
- 06.05.2005 zweiter Preis und Preis für Zubehör von Jens Schmietendorf, Beispiel Template von http://bananaguard.de , zk_products 1.3.2 von Klaus Zierer, MWSt von Franz Holzinger
- 23.06.2005 products_mail.inc gelöscht, Beispiel Template von Franz Holzinger
- 26.07.2005 Anzeige der Bestellungen, Gutscheinpunkte System und Freundschaftswerbung von Bert Hiddink
- 03.08.2005 Anmerkungen aus dem zk_products Forum, PIDtracking
- 11.09.2005 Geschenk Gutscheine -14.09.2005 Flexforms statt CODEs
- 05.10.2005 Ergänzungen von Andrea Herzog-Kienast (--> Übersetzer für das französische Manual gesucht)
- 13.10.2005 Gutscheinpunkte und Freundschaftswerbung
- 28.10.2005 Zubehör ist durch Größe mit Eintrag in der Artikeltabelle ersetzt worden.
- 14.07.2006 Ab jetzt muß das statische Shop-Template manuell im Template Setup eingebunden werden.
  CSS-Shop Template von Robert Markula für ein barrierefreies Shop